



UNIVERSIDAD POLITÉCNICA DE MADRID  
FACULTAD DE INFORMÁTICA

TRABAJO FIN DE CARRERA

USABILITY PLANNER  
HERRAMIENTA PARA LA PLANIFICACIÓN DE TÉCNICAS  
DE USABILIDAD

AUTOR: Tomás Antón Escobar

TUTOR: Xavier Ferré Grau



Este documento se distribuye bajo licencia Creative Commons Reconocimiento – No comercial – Sin obras derivadas 3.0 Unported. Para ver una copia de esta licencia visite: <http://creativecommons.org/licenses/by-nc-nd/3.0/>

*In effect, that everything should be as  
simple as possible, but not simpler.*

(En efecto, todo debería ser tan simple  
como sea posible, pero no más simple).

Albert Einstein





# ÍNDICE

<b>1. Introducción. ....</b>	<b>1</b>
1.1 Alcance y objetivos. ....	2
1.2 Descripción de la aplicación. ....	5
1.3 Herramientas anteriores. ....	12
1.4 Estructura de este documento. ....	13
<b>2. Selección de métodos de usabilidad en el ciclo de vida. ....</b>	<b>15</b>
2.1 La usabilidad en el ciclo de vida del software. ....	16
2.2 El Diseño Centrado en el Usuario. ....	20
2.2.1 Elementos del modelo. ....	22
2.3 Técnicas de usabilidad para el soporte del proceso DCU. ....	30
2.3.1 Clasificación del ISO/TR 16982. ....	30
2.3.2 Clasificación del ISO/WD 9241-230. ....	35
2.3.3 Clasificación por etapas del ciclo de vida. ....	36
2.3.4 ¿En qué etapa del ciclo de vida se debe elaborar el plan de usabilidad? ....	43
<b>3. Diseño de la interacción. ....</b>	<b>44</b>
3.1 Primera iteración. ....	45
3.1.1 Conceptualización. ....	45
3.1.2 Comprendiendo necesidades y capturando requisitos. ....	50
3.1.3 Creación de Personas. ....	51
3.1.4 Análisis de tareas. ....	55
3.1.5 Nombre de la aplicación. ....	57
3.1.6 Wireframes. ....	59
3.1.7 Desarrollo del prototipo. ....	61
3.2 Segunda iteración. ....	62
3.2.1 Revisión de la conceptualización. ....	62
3.2.2 Revisión de las personas y análisis de tareas. ....	66
3.2.3 Creación de un escenario. ....	67
3.2.3 Revisión del nombre de la aplicación. ....	69
3.2.4 Producción de una nueva solución de diseño. ....	70
3.2.5 Realización de un test de usabilidad. ....	79
3.2.5 Desarrollo de la solución de diseño. ....	81
3.3 Tercera iteración. ....	87
3.3.1 Cambios a nivel de diseño gráfico. ....	88
3.3.2 Ajustes en el prototipo. ....	89

3.3.2 Diseño del póster y creación de un artículo.....	93
3.3.3 Evaluación del prototipo durante el congreso. ....	95
3.4 Cuarta iteración. ....	97
3.4.1 Inclusión de una nueva Persona. ....	98
3.4.2 Creación de un segundo escenario. ....	99
3.4.3 Revisión de las tareas. ....	100
3.4.4 Producción de una nueva solución de diseño. ....	101
3.4.5 Evaluación de la versión alcanzada. ....	103
<b>4. Estudio de tecnologías web para la implementación. ....</b>	<b>107</b>
4.1 Aplicaciones web de primera y segunda generación. ....	108
4.2 Aplicaciones web de tercera generación. ....	109
4.3 Aplicaciones web de cuarta generación. ....	109
4.3.1 Tecnologías que implementan la web 2.0. ....	110
4.3.2 Aplicaciones RIA.....	112
4.4 Soluciones elegidas para la implementación de Usability Planner. ....	116
4.5 GWT (Google Web Toolkit).....	118
4.6 XML.....	121
<b>5. Desarrollo del sistema.....</b>	<b>123</b>
5.1 Capas arquitectónicas.....	124
5.1.1 Interfaz de usuario. ....	125
5.1.2 Lógica de negocio. ....	126
5.1.3 Capa de servicios.....	126
5.2 El patrón MVP. ....	127
5.2.1 Funcionamiento del presentador. ....	129
5.2.2 MVP vs MVC. ....	129
5.3 Diagrama de clases.....	130
5.3.1 Diagrama de clases (notación de paquetes).....	131
5.3.2 Diagrama de clases del paquete modelo. ....	133
5.3.3 Diagrama de clases del paquete presentadores. ....	137
5.3.4 Diagrama de clases del paquete vistas. ....	140
5.4 Diseño de la base de datos.....	142
<b>6. Conclusiones y líneas futuras.....</b>	<b>144</b>
6.1 Conclusiones.....	145
6.2 Líneas futuras.....	149
<b>Bibliografía .....</b>	<b>153</b>
<b>Anexos .....</b>	<b>157</b>

A1. Relación entre buenas prácticas, técnicas y etapas.....	158
A2. Etapas del ciclo de vida consideradas. ....	162
A.2.1 Actividades o etapas del ciclo de vida del Enfoque DCU .....	163
A.2.2 Etapas del ciclo de vida del Proceso Unificado de Rational.....	164
A3. Cuestionario de satisfacción.....	165



## Capítulo 1.

# INTRODUCCIÓN.

A pesar del creciente interés de las organizaciones por la aplicación de métodos y técnicas de usabilidad en el proceso de desarrollo de software, seleccionar y poner en práctica aquellas técnicas que más se adecuan a las características de un determinado proyecto es una tarea compleja y con un soporte bajo.

La diversidad de métodos de usabilidad, su desconocimiento en el mundo profesional y la falta de unas pautas claras para su aplicación en un determinado proyecto están causando que la introducción del Diseño Centrado en el Usuario (UCD, del inglés *User Centred Design* o DCU en español) en las compañías sea una tarea costosa y compleja.

No obstante, las empresas que apuestan por la usabilidad de sus productos se están posicionando por encima del resto al crear sistemas que los usuarios valoran notablemente y por ver incrementadas sus ventas.

Jefes de proyecto y consultores, que ven atractivo el desarrollo de sistemas usables, se hallan en un escenario en el que no encuentran un mecanismo sencillo e inmediato para seleccionar y aplicar las técnicas de usabilidad en sus proyectos de forma ágil y estructurada. El criterio para la selección de técnicas y métodos que manejan se basa únicamente en su experiencia profesional y en la bibliografía de numerosos autores.

Usability Planner es una aplicación web dirigida a profesionales, estudiantes e investigadores involucrados en el desarrollo de software para ayudar a cambiar este escenario. Su objetivo es dar soporte a la selección de técnicas y métodos de usabilidad en el proceso de desarrollo de software minimizando riesgos y maximizando beneficios.

## 1.1 Alcance y objetivos.

Para las empresas la meta es crear, en el menor tiempo posible, productos exitosos que cumplan todos los requisitos invirtiendo los menos recursos posibles y minimizando los riesgos. En un entorno corporativo en donde la usabilidad se concibe como causa de éxito, uno de los factores principales para lograr esta meta es conocer los métodos y técnicas más efectivos para introducir mejoras en la usabilidad de los productos (Wixon, 2003).

Este planteamiento arroja la importancia de saber elegir bien los métodos y técnicas de usabilidad que cada proyecto requiere. Investigadores y profesionales de la usabilidad han ido con los años perfilando procesos, técnicas y métodos para construir software de calidad a medida de las necesidades de los usuarios. El problema es que dicha información, aunque abundante, aún no está claramente estructurada y es complejo encontrar referencias claras, prácticas y concisas que guíen a los profesionales en la selección de técnicas y métodos de usabilidad dentro del ciclo de vida de sus desarrollos.

Así lo menciona Seffah y Metzker, “Las técnicas y métodos de usabilidad son todavía poco utilizados y difícilmente comprensibles para equipos de desarrollo de software y organizaciones (Seffah & Metzker, 2004).”

Un claro ejemplo es la diversidad de estándares internacionales que en materia de usabilidad, ergonomía e interacción persona-ordenador se han redactado en los últimos 25 años:

- El ISO 20282-1:2006 provee recomendaciones para el diseño de sistemas fáciles de manejar atendiendo al contexto de uso del usuario.
- El ISO 9126-1 es un estándar para la evaluación de la calidad del software. Establece las características que un sistema informático de calidad debe cumplir, entre ellas la usabilidad.
- El ISO 9241-11 proporciona una guía sobre la usabilidad en sistemas basados en terminales visuales.
- El ISO 25062:2006 especifica un marco para la evaluación de la usabilidad de un sistema software.

- El ISO 13407:1999 describe el proceso DCU para sistemas interactivos. Este estándar ha sido actualizado y renombrado como ISO 9241-210.
- El ISO TR 18529 desglosa cada actividad del proceso DCU y proporciona una base para aquellos que deseen incluir actividades, técnicas y métodos de usabilidad en su proyecto.
- El ISO PAS 18152 proporciona un modelo para la creación de sistemas usables, saludables y seguros.

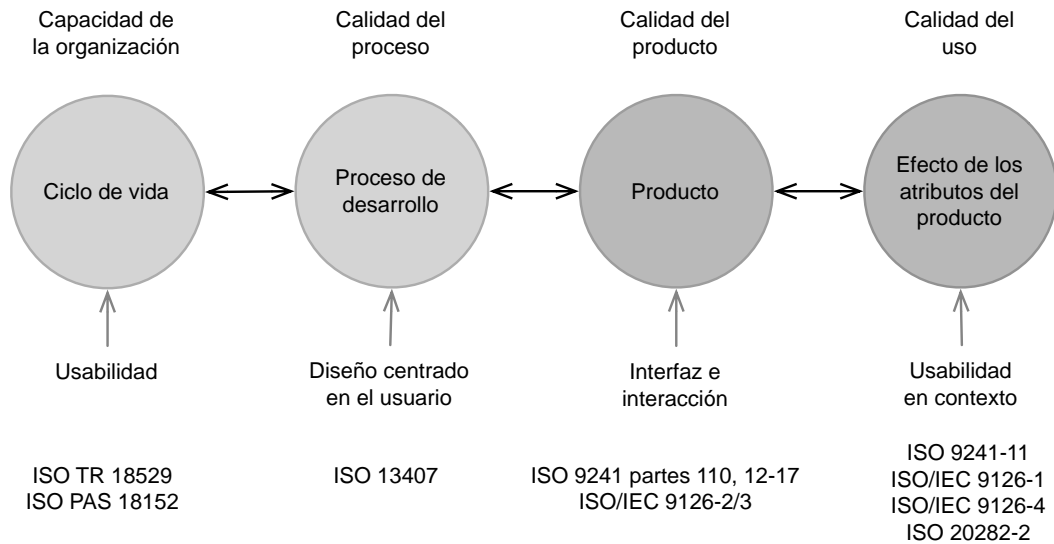


Fig. 1. Los estándares sobre usabilidad (Bevan, 2007).

Uno de los principales objetivos de los estándares es aportar consistencia, compatibilidad y seguridad sobre la disciplina que tratan, por lo que deberían ser la primera referencia para un profesional dedicado a la usabilidad a la hora de seleccionar una fuente de información.

La realidad es que la mayoría de profesionales de la usabilidad no utilizan los estándares. Una de las razones puede ser el coste, por ejemplo el ISO 13407 con tan solo 20 páginas cuesta 110 dólares.

Ante este escenario parece necesario acercar a profesionales, estudiantes e investigadores estos estándares de forma que puedan ser llevados a la práctica.

En este sentido el Dr. Nigel Bevan, profesional con más de 15 años de experiencia en consultoría de usabilidad, publicó en la segunda edición del I-USED (Taller Internacional de Desarrollo de Software y Evaluación de Usabilidad), dentro del INTERACT 2009 (Congreso Internacional de Interacción Persona-Ordenador), un artículo (Bevan, 2009) en el que pone de manifiesto la complejidad del procedimiento actual de selección de métodos de usabilidad sugerido por el ISO TR 16982 y por el ISO PAS 18152.

Bevan propone un mecanismo paso a paso para la selección de métodos y técnicas de usabilidad, tomando como base los dos informes técnicos anteriores, en función de las limitaciones de un determinado proyecto minimizando los riesgos y maximizando los beneficios.

Por otro lado, son cada día más los desarrolladores que toman conciencia de la importancia de incluir en el proceso de creación de software métodos y técnicas de usabilidad y no encuentran una referencia clara y dirigida específicamente a ellos. Por ejemplo, para los desarrolladores, técnicas de usabilidad como los *wireframes* (bocetos del sistema, en español) son considerados una excelente técnica de usabilidad ya que les aporta de forma rápida, visual e interactiva todo lo que necesitan para la implementación del sistema, desde el diseño y arquitectura de la interfaz a la navegación y los casos de uso.

Este perfil suele tomar como referencias para sus proyectos la bibliografía de numerosos autores con gran experiencia en usabilidad e interacción persona-ordenador. En este sentido el Dr. Xavier Ferré, profesor de Interacción Persona-Ordenador e Ingeniería del Software, con más de 15 años de experiencia, ha identificado las técnicas de usabilidad más recomendadas a aplicar en las distintas etapas del ciclo de vida del software en un contexto dirigido a desarrolladores (Ferré, 2005). En su tesis identificó nada menos que 96 categorías diferentes de métodos y técnicas de usabilidad.

La cifra anterior es inquietante y puede ser uno de los motivos por los que perfiles no ligados al ámbito de la usabilidad vean ardua la tarea de seleccionar métodos y técnicas de usabilidad.



Ante este problema se plantea la necesidad de desarrollar una aplicación que resuelva la complejidad de saber qué métodos y técnicas de usabilidad utilizar en un contexto organizacional, maximizando los beneficios y minimizando los riesgos.

Usability Planner es el resultado, una herramienta *online* disponible en la dirección: [www.usabilityplanner.org](http://www.usabilityplanner.org) que incluye las recomendaciones del ISO TR 16982, extendidas con la experiencia de Nigel Bevan y Xavier Ferré en la selección de las técnicas de usabilidad más apropiadas para un proyecto.

## 1.2 Descripción de la aplicación.

Usability Planner clasifica los métodos y técnicas de usabilidad recomendadas en cada etapa del ciclo de vida en función de una serie de limitaciones asociadas a un determinado proyecto (basadas en el ISO PAS 18152), en un determinado contexto organizacional. Además da soporte a la priorización de las técnicas minimizando riesgos y maximizando beneficios.

Permite a profesionales, investigadores, estudiantes y desarrolladores elaborar un **plan de usabilidad** en función de las etapas del ciclo de vida en las que consideren que la usabilidad puede minimizar los riesgos y maximizar beneficios en función de las limitaciones asociadas a un determinado proyecto.

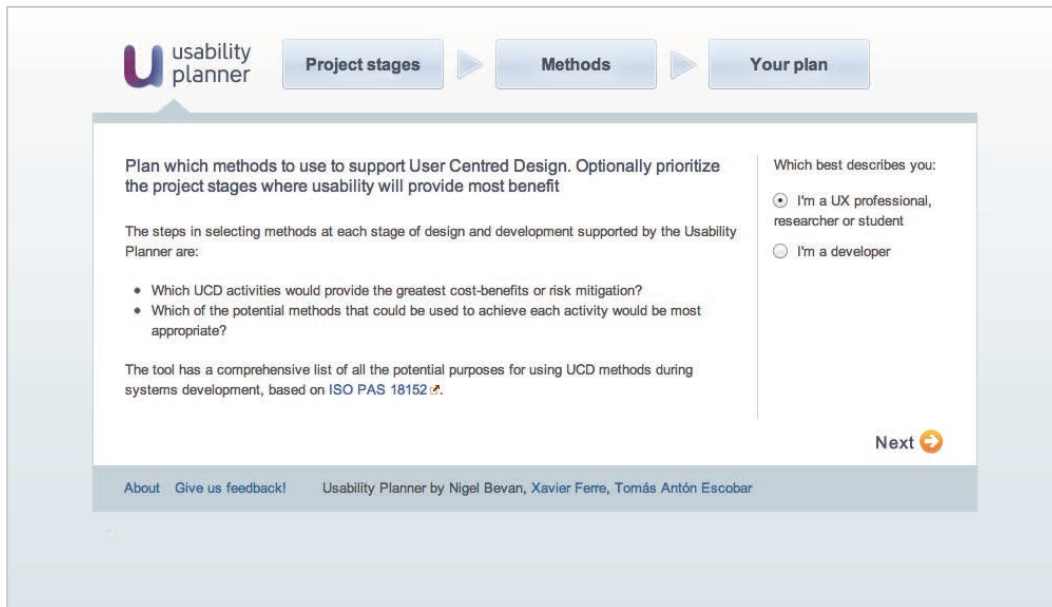


Fig. 2. Pantalla inicial de Usability Planner.

Usability Planner se basa en dos ideas fundamentales:

- ¿Qué etapas del ciclo de vida son las más beneficiadas por el uso de técnicas y métodos de usabilidad?
- En cada una de las etapas en las que se van a aplicar técnicas y métodos de usabilidad ¿Cuáles son los métodos más apropiados?

Responder a las dos preguntas anteriores es sencillo con Usability Planner. La priorización de técnicas y métodos de usabilidad se obtiene con tan sólo tres pasos:

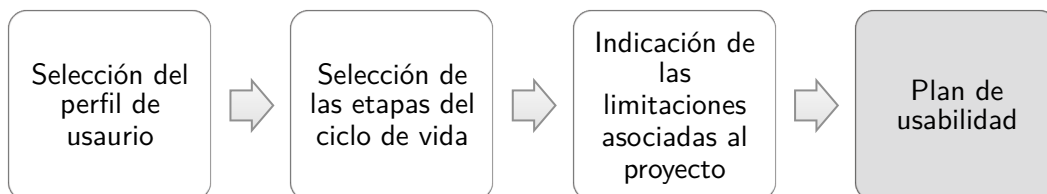
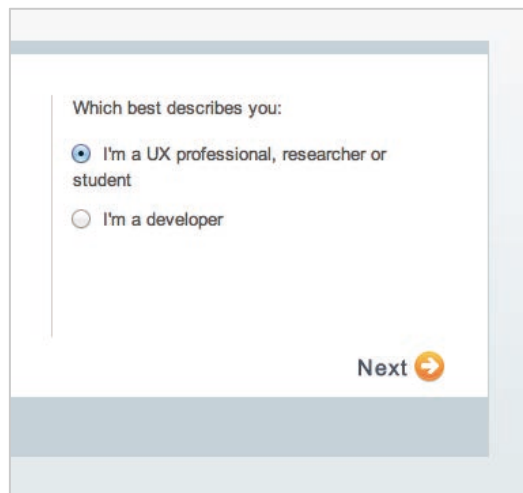


Fig. 3 Pasos para la obtención del plan de usabilidad.

1. Primero el usuario debe indicar el perfil que mejor le describa (ver Fig. 4). La aplicación permite elegir entre dos tipos de perfiles: Estudiantes, profesionales o investigadores en el ámbito de la usabilidad o bien desarrolladores. Esta selección determina las etapas del ciclo de vida en las que la aplicación de técnicas y métodos de usabilidad minimizará riesgos.
  - a. Para el perfil de estudiantes, profesionales o investigadores en el área de la usabilidad se ofrecen las actividades incluidas en el ISO PAS 18152.
  - b. Para el perfil de desarrolladores se ofrecen las actividades del Proceso Unificado (Jacobson, Booch, & Rumbaugh, 1999).



Which best describes you:

☒ I'm a UX professional, researcher or student

☐ I'm a developer

Next →

Fig. 4 Selección del perfil de usuario.

2. A continuación, el usuario debe identificar las etapas del ciclo de vida en las que prevé aplicar técnicas de usabilidad (ver Fig. 5). Las etapas se ofrecen organizadas en subetapas (si aplica) permitiéndose la selección de subetapas por separado.

Además los usuarios pueden consultar una breve descripción de cada una de las etapas, así como las buenas prácticas que, en materia de usabilidad, se recomiendan aplicar y que son obtenidas por el empleo de los métodos y técnicas que la aplicación recomienda.

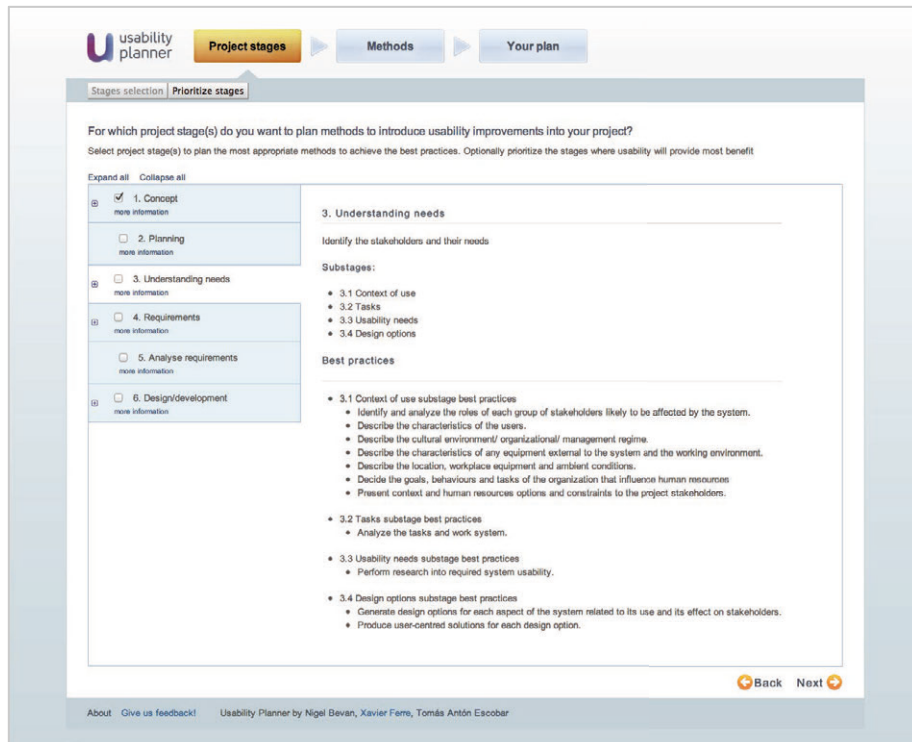


Fig. 5. Selección de etapas del ciclo de vida del proyecto.

Adicionalmente los usuarios pueden priorizar las etapas elegidas en función de restricciones asociadas a beneficios o riesgos de negocio, tales como incrementar la productividad, reducir costes de mantenimiento y soporte, etc. Esta selección es simplemente una ayuda para que los usuarios tomen conciencia de cuáles son las etapas más prioritarias. En la versión actual de la aplicación, la selección de estas restricciones no interfiere en el resultado final.

3. En la sección *Methods* (métodos en español), el usuario puede indicar las limitaciones asociadas a su proyecto y su contexto, en términos de restricciones como el tiempo, costes, experiencia, acceso a los usuarios, naturaleza de las tareas a realizar, complejidad, contexto de uso, etc. (ver Fig. 6).

La selección puede llevarse a cabo etapa a etapa o de forma global a todas las etapas del proyecto (ver Fig. 7). Usability Planner, de la lista de métodos y técnicas anterior, ofrece y prioriza sólo aquellas que minimicen los riesgos y maximicen los beneficios de negocio.

usability planner

Project stages Methods Your plan

All stages selection Stage by stage selection Overview selection

How cost effective is each possible method likely to be?

Specify the constraints that will influence which UCD methods are appropriate in your situation

**Project constraints**

- ☒ Need quick results
- ☒ Very restricted budget
- ☐ Usability important
- ☐ Uncertain specification

**User constraints**

- ☐ Difficult to involve users
- ☐ No access to users
- ☐ Some users have disabilities
- ☐ Mostly first time users

**Task constraints**

- ☒ Complex task
- ☐ Many tasks
- ☐ Safety or business critical system
- ☐ Organizational changes needed

**Product constraints**

- ☐ Efficiency or accuracy is important
- ☐ Adaptation of an existing system
- ☐ A well understood product
- ☐ Customisable product

**Context of use constraints**

- ☐ General purpose: used in many different contexts

**Human constraints**

- ☐ No usability expertise available

**1. Concept / 1.1 Envisioning opportunities**

- ☒ Braindrawing ★★★☆☆
- ☒ Brainstorming ★★★☆☆
- ☒ Preliminary field visit ★★★☆☆
- ☒ Photo study ★★☆☆☆
- ☒ Focus groups ☆☆☆☆☆
- ☒ Simulations of future working environments ☆☆☆☆☆

**1. Concept / 1.2 System scoping**

- ☒ Consult stakeholders ★★★☆☆
- ☒ Context of use analysis ★★★☆☆
- ☒ Participatory workshops ★★☆☆☆

**2. Planning**

- ☒ Plan to achieve and maintain usability ☆☆☆☆☆

Back Next

About Give us feedback! Usability Planner by Nigel Bevan, Xavier Forre, Tomás Antón Escobar

Fig. 6. Selección de las limitaciones asociadas al proyecto.

La priorización de los métodos y técnicas se realiza en tiempo real, de forma que el usuario puede ver cómo afecta la introducción o eliminación de una determinada limitación.

usability planner

Project stages ▶ **Methods** ▶ Your plan

All stages selection | Stage by stage selection | Overview selection

How cost effective is each possible method likely to be?  
Specify the constraints that will influence which UCD methods are appropriate in your situation

All constraints | Project | User | Task | Product | Context of use | Human

	1. Concept	1.1 Envisioning opportunities	1.2 System scoping	2. Planning	
<b>Project constraints</b>					<b>Project constraints</b>
<input type="checkbox"/> Need quick results	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Need quick results
<input type="checkbox"/> Very restricted budget	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Very restricted budget
<input type="checkbox"/> Usability important	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Usability important
<input type="checkbox"/> Uncertain specification	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Uncertain specification
<b>User constraints</b>					<b>User constraints</b>
<input type="checkbox"/> Difficult to involve users	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Difficult to involve users
<input type="checkbox"/> No access to users	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> No access to users
<input type="checkbox"/> Some users have disabilities	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Some users have disabilities
<input type="checkbox"/> Mostly first time users	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Mostly first time users
<b>Task constraints</b>					<b>Task constraints</b>
<input type="checkbox"/> Complex task	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Complex task
<input type="checkbox"/> Many tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Many tasks
<input type="checkbox"/> Safety or business critical system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Safety or business critical system
<input type="checkbox"/> Organisational changes needed	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Organisational changes needed
<b>Product constraints</b>					<b>Product constraints</b>
<input type="checkbox"/> Efficiency or accuracy is important	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Efficiency or accuracy is important
<input type="checkbox"/> Adaptation of an existing system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Adaptation of an existing system
<input type="checkbox"/> A well understood product	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> A well understood product
<input type="checkbox"/> Customisable product	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Customisable product
<b>Context of use constraints</b>					<b>Context of use constraints</b>
<input type="checkbox"/> General purpose: used in many different contexts	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> General purpose: used in many different contexts
<b>Human constraints</b>					<b>Human constraints</b>
<input type="checkbox"/> No usability expertise available	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> No usability expertise available

Back Next

About Give us feedback! Usability Planner by Nigel Bevan, Xavier Feme, Tomás Antón Escobar

Fig. 7. Selección de limitaciones (vista general).

En tan sólo tres pasos Usability Planner elabora un plan (ver Fig. 8) para introducir las técnicas y métodos de usabilidad más recomendados según los criterios indicados por el usuario en el ciclo de vida de su proyecto.

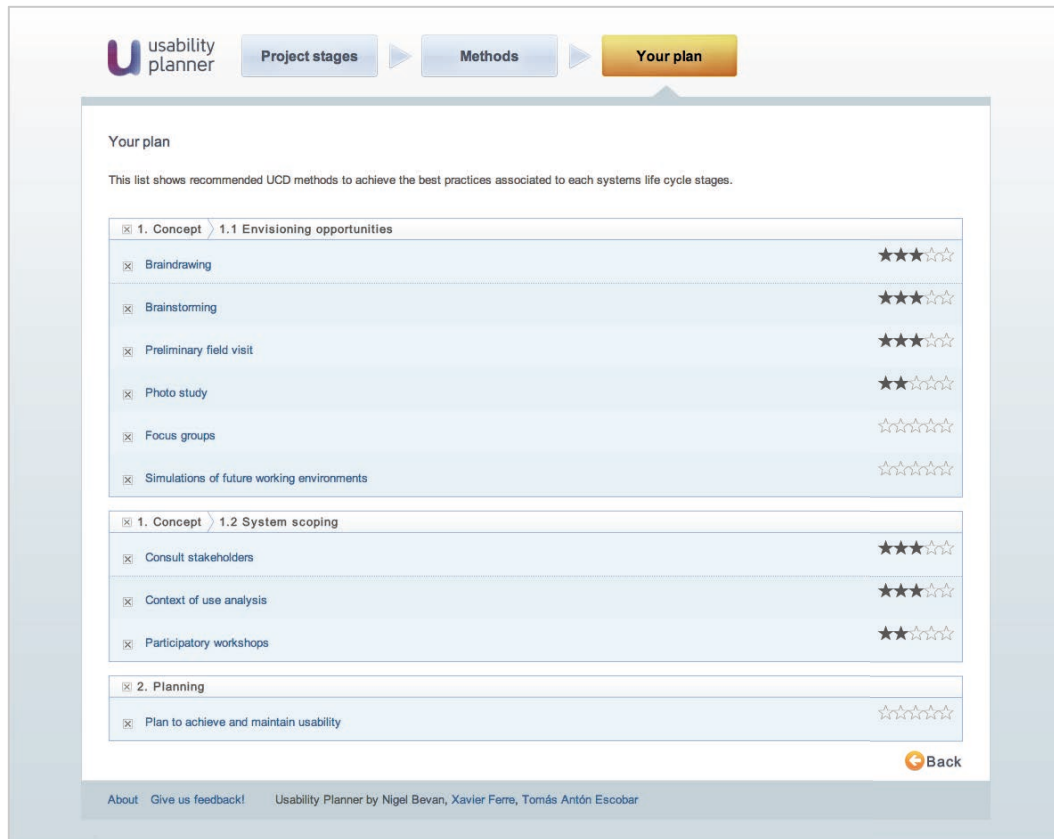


Fig. 8. Plan con la priorización de las técnicas y métodos.

La aplicación además da soporte para que profesionales experimentados en usabilidad, que conocen los métodos y técnicas, puedan ajustar el plan eliminando aquellas técnicas o métodos que consideren que no aplican en su proyecto o en la organización en la que se encuentran por no ser viable su aplicación.

Si el usuario descarga la aplicación puede utilizarla sin conexión e incluso editar la base de datos para personalizar los resultados o adaptarlos a su contexto organizacional. Esto es especialmente útil para organizaciones que ya han asumido la introducción de técnicas y métodos de usabilidad en sus procesos y que conocen las técnicas que mayores beneficios les aportan en su contexto. En este sentido la aplicación es escalable y personalizable.

Usability Planner ha sido desarrollado como proyecto de código abierto. Cualquier usuario u organización puede tener acceso al código fuente pudiendo ajustarla y personalizarla en función de sus necesidades.

Se distribuye bajo la licencia Creative Commons con atribución, uso no comercial y modificación siempre que el resultado conserve la misma licencia (Creative Commons, 2012).

La aplicación ha sido desarrollada siguiendo el proceso DCU, mediante 4 iteraciones incrementales. En sucesivos capítulos se detalla ampliamente como ha sido este proceso.

## 1.3 Herramientas anteriores.

A pesar de la extensa literatura que trata y describe diferentes técnicas y métodos de usabilidad, muy poca trata el problema de seleccionar las técnicas y métodos más apropiados.

Uno de los primeros intentos fue el sistema experto INTUIT, desarrollado dentro del proyecto HUFIT (Galer & Russell, 1987) el cual recomendaba qué métodos de usabilidad se debían emplear.

Más recientemente, otros sistemas como UsabilityNet han categorizado los métodos basándose en criterios como los recursos, el acceso a los usuarios y la experiencia o habilidades del equipo.

La solución más sofisticada es la recogida en el estándar ISO TR 16982, el cual sugiere un criterio para la selección de tipos de métodos ofreciendo una recomendación en función de limitaciones asociadas al proyecto. El problema de esta aproximación es que su aplicación o puesta en práctica es costosa. El usuario tiene que navegar por varias tablas y extraer de forma manual la priorización de los tipos de métodos. Y el mero hecho de ofrecer tipos de métodos y no los métodos en sí dificulta aún más la selección.

Lo deseable sería ofrecer una herramienta más ágil en la que el usuario no tuviera que conocer y preocuparse de la selección de los métodos, si no de las características asociadas a su proyecto y organización que pueden hacer más recomendadas o menos unos métodos u otros.



## 1.4 Estructura de este documento.

Este documento se organiza en 6 capítulos y 3 anexos:

- **Primer capítulo: Introducción.** El presente capítulo ofrece el alcance, objetivos y descripción del proyecto realizado y sirve de punto de partida para introducir el resto de capítulos.
- **Segundo capítulo: Selección de métodos de usabilidad en el ciclo de vida.** Describe el fundamento teórico en el que se basa la herramienta desarrollada. En este capítulo se presentan diferentes criterios para la selección de técnicas y métodos de usabilidad.
- **Tercer capítulo: Diseño de la interacción.** Este capítulo describe cómo ha sido el proceso de Diseño Centrado en el Usuario de la aplicación, desde su conceptualización hasta su puesta en producción, junto con las técnicas y métodos que se han empleado para garantizar la usabilidad.
- **Cuarto capítulo: Soluciones tecnológicas para la implementación de Usability Planner.** Se comienza realizando un breve recorrido por las distintas tecnologías que desde los orígenes de la web han dado soporte al desarrollo de aplicaciones. Una vez identificadas, se justifica la solución tecnológica adoptada para dar soporte a Usability Planner.
- **Quinto capítulo: Desarrollo del sistema.** En este capítulo se describe cómo se ha llevado a cabo la implementación del sistema. Se presenta su arquitectura y se explica cada uno de los componentes que la integran.
- **Sexto capítulo: Conclusiones y líneas futuras.** Por último, se exponen las conclusiones a las que se han llegado tras la realización del proyecto, así como, posibles líneas futuras de ampliación y mejora.
- **Anexo 1: Relación entre buenas prácticas, técnicas y etapas.** Este anexo recoge las distintas técnicas y las buenas prácticas identificadas por el ISO PAS 18152.
- **Anexo 2. Etapas del ciclo de vida consideradas.** Este anexo describe las etapas del ciclo de vida que se manejan en Usability Planner para perfiles de usuario ligados profesionalmente a la usabilidad (profesionales, estudiantes e investigadores) y para perfiles más ligados al desarrollo.

- **Anexo 3. Ejemplo de cuestionario de satisfacción.** Este anexo incluye el modelo de cuestionario usado en el congreso NordiCHI de 2010 para recoger la opinión de los usuarios participantes.

## Capítulo 2.

# SELECCIÓN DE MÉTODOS DE USABILIDAD EN CICLO DE VIDA.

La factorización del software y la falta de interés por parte de las organizaciones por los usuarios a los que dirigen sus productos, han propiciado un escenario en el que la usabilidad emerge como un atributo de éxito.

La demanda de software de calidad, dirigido a satisfacer las necesidades de los usuarios, está demostrando que, con un poco más de inversión, se pueden obtener productos de éxito, usables, con un elevado ROI (Retorno de la inversión, del inglés *Return Of Investment*).

En contraposición al diseño de software tradicional, dónde el rendimiento y la fiabilidad es lo más importante y la usabilidad se liga, generalmente, a la etapa de diseño de la interfaz de usuario, el enfoque DCU se ocupa de introducir métodos y técnicas de usabilidad en el ciclo de vida de un sistema para lograr un resultado a medida de los usuarios.

Dentro de este proceso la selección de métodos y técnicas de usabilidad no es a día de hoy una tarea sencilla (Seffah & Metzker, 2004). En el capítulo anterior se avanzó el problema de encontrar referencias claras que permitan la selección de métodos y técnicas de usabilidad de forma simple. Así los profesionales tienen dos fuentes principales para encontrar respuesta a este problema, la extensa bibliografía creada por expertos en usabilidad y los estándares internacionales.

En este capítulo se presentan diferentes criterios para la selección de técnicas y métodos de usabilidad para llevar a cabo el enfoque DCU, en concreto los que dan soporte a Usability Planner.

## 2.1 La usabilidad en el ciclo de vida del software.

La Ingeniería del Software se caracteriza por aplicar un enfoque sistemático, disciplinado y cuantificable al desarrollo, ejecución y mantenimiento del software (SWEBOK, 2004).

Al igual que en otras disciplinas, las compañías de desarrollo siempre se han preocupado por definir una metodología o proceso que les permita construir software de calidad minimizando riesgos y que les reporte los máximos beneficios. Automatizar tareas, facilitar el trabajo del equipo, mejorar la gestión, el soporte e incrementar la calidad de los entregables, de forma constante, son atributos deseables de todo proceso o modelo de desarrollo.

En el contexto de la Ingeniería del Software la usabilidad se entiende como un atributo de calidad, ligado principalmente a la interfaz de usuario, pero desligado del proceso y entendido como un plus adicional para construir sistemas mejores.

El ISO 9241-11 define la usabilidad como una meta:

*“Capacidad de un producto de poder ser usado por unos usuarios específicos para alcanzar sus metas de forma eficaz, con eficiencia y satisfacción en un determinado contexto de uso”.*

La definición anterior pone de manifiesto que la usabilidad no es un atributo inherente al software, es una característica intrínseca que va más allá del aspecto de la interfaz de usuario y que cumple el software útil, sencillo, que produce satisfacción y que gusta a un amplio conjunto de usuarios, en un determinado contexto de uso.

La usabilidad además no puede especificarse independientemente del entorno de uso y de los usuarios concretos que vayan a utilizar el sistema (Ferré, 2000). El usuario es el destinatario final del software y el uso que él haga del mismo, en un determinado contexto, es una de las cuestiones más importantes a plantear a lo largo del proceso.

La usabilidad es más que un atributo de calidad, cuando un usuario tiene una buena experiencia al utilizar un sistema se dice que dicho sistema tiene un nivel aceptable de usabilidad. Un sistema con un nivel aceptable de usabilidad es eficaz, eficiente, fácil de aprender y de recordar y transmite al usuario tranquilidad en lugar de ansiedad.

En toda organización se buscan mecanismos automáticos y factorizables para incrementar la productividad y el rendimiento. En Ingeniería del Software también. Sin embargo, la usabilidad no se puede factorizar, principalmente porque la mayoría de las técnicas requieren la participación activa de los usuarios y esto genera incertidumbre.

Otra barrera a la que se enfrenta la usabilidad son los modelos de ciclo de vida que ha dado lugar la Ingeniería del Software. Desde que en 1970 se estableciera el ciclo de vida en cascada hasta hoy, se han ido proponiendo otros modelos, no llegándose a consolidar ninguno de ellos.

En el ciclo de vida en cascada (ver Fig. 9), constituido por etapas que se realizan en orden, de forma que el inicio de una etapa debe esperar a la finalización de la anterior (S. Presuman, 2005), la usabilidad está frecuentemente ligada a la etapa de diseño, en términos de disposición, diseño y organización de la interfaz gráfica de usuario.

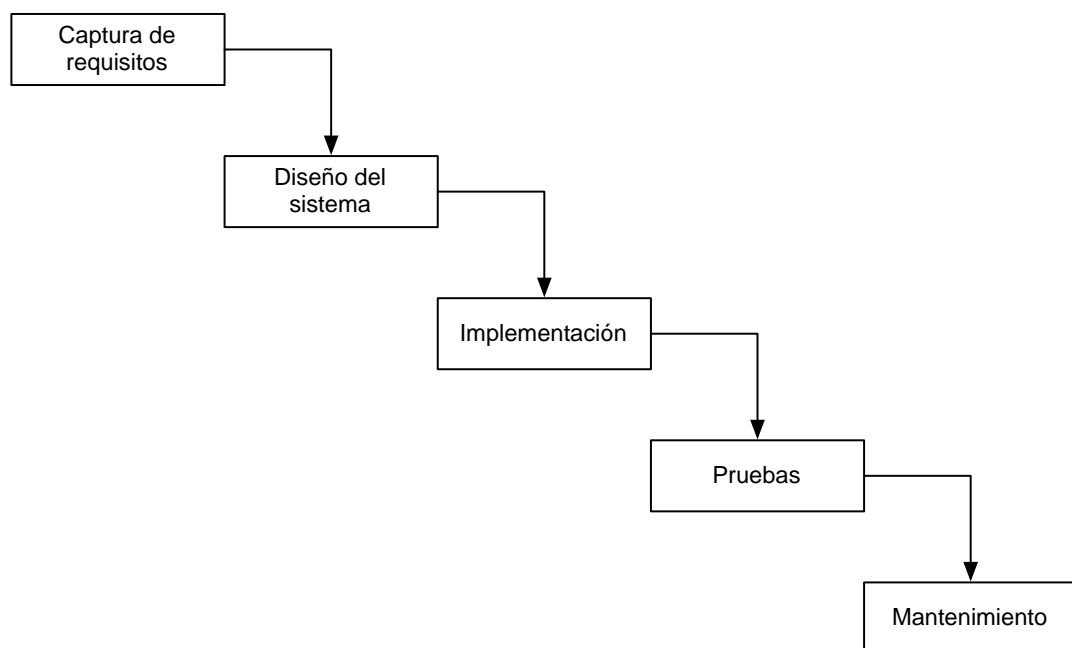


Fig. 9. Etapas del ciclo de vida en cascada.

Este tipo de modelo no favorece la usabilidad, principalmente porque los requisitos y las necesidades de los usuarios pueden cambiar durante el ciclo de vida y porque la usabilidad no puede condensarse sólo en una etapa del modelo, ya que está ligada a la interacción del usuario con el sistema y esta interacción queda definida en cada una de las etapas.

La usabilidad requiere de flexibilidad y adecuación al cambio dentro del proceso, por ello encaja mejor en escenarios iterativos e incrementales (Fig. 10). El desarrollo de un sistema de forma incremental, mejorando y ampliando el mismo mediante versiones o entregables es un buen escenario para la inclusión de métodos y técnicas de usabilidad.

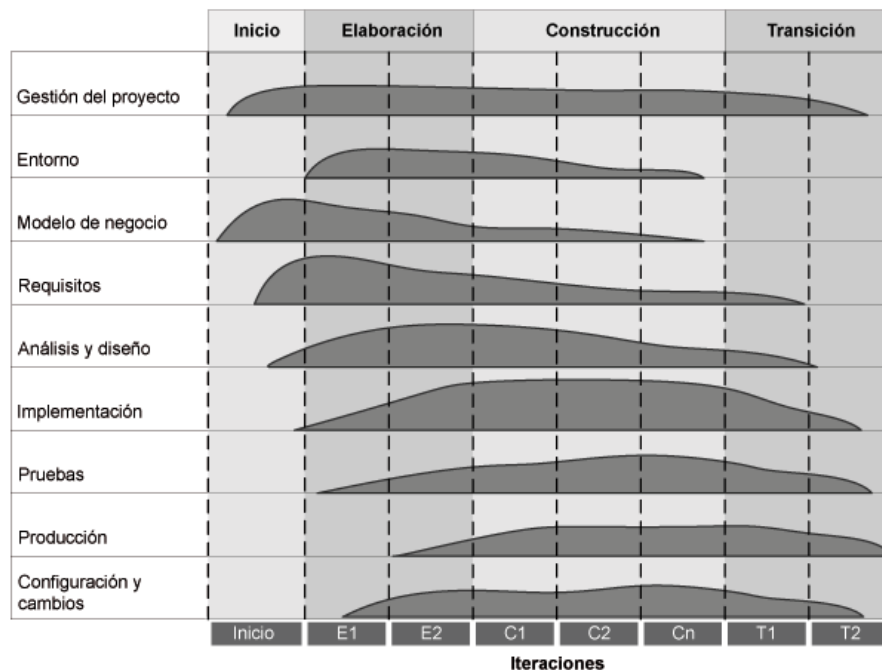


Fig. 10. Ejemplo de ciclo de vida iterativo e incremental.

La reciente inclusión en la Ingeniería del Software de las llamadas metodologías ágiles favorece un escenario en el que la usabilidad tiene más cabida, ya que se caracterizan por no adoptar un ciclo de vida concreto (aunque cada metodología tiene el suyo) y por incluir una rápida respuesta al cambio, ampliar la comunicación e incluir, en muchos casos, la opinión de los usuarios.

No obstante, se necesita un modelo en el que el usuario sea realmente el centro del proceso, estando presente en todas las etapas del ciclo de vida y sea verdaderamente el foco en la toma de decisiones.

En este sentido, algunos modelos se han propuesto desde la disciplina HCI (del inglés *Human Computer Interaction*, Interacción Persona Ordenador en español), la cual estudia la comunicación e interacción de las personas con sistemas interactivos, con el fin de obtener sistemas fáciles de usar y de entender, que generen satisfacción al usuario.

Uno de los primeros modelos propuestos fue el ciclo de vida en estrella (ver Fig. 11) de Hartson y Hix (Hartson & Hix, 1989), el cual identifica dos grupos de actividades, uno con enfoque *top-down* (de arriba hacia abajo) desde la perspectiva del sistema hacia la del usuario y otro con enfoque *bottom-up* (de abajo hacia arriba), desde la perspectiva del usuario hacia la del sistema. Se caracteriza por no establecer un orden de aplicación de cada una de sus actividades, de hecho todas están interconectadas, pudiéndose mover por cada una de ellas pero pasando siempre por la evaluación, la cual es el centro de este modelo.

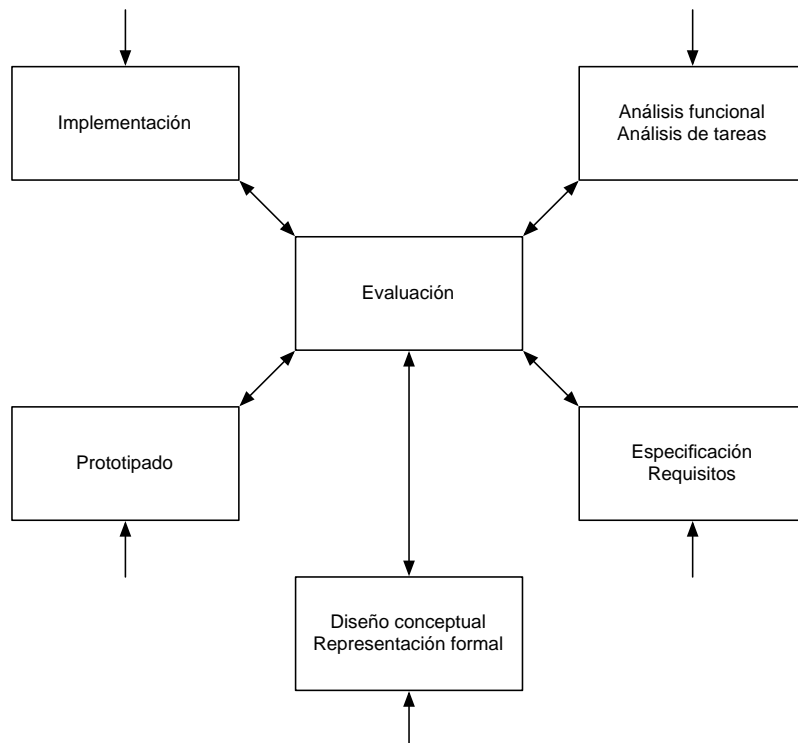


Fig. 11. Ciclo de vida en estrella.

Posteriormente al ciclo de vida en estrella se publicó el estándar ISO 13407 en el que se define el enfoque DCU. De todos es el que más se aproxima al objetivo de crear software que resuelva las necesidades concretas de los usuarios maximizando la usabilidad y la experiencia de usuario.

La principal diferencia con respecto a otros modelos establecidos por la Ingeniería del Software es que la consecución de un sistema de calidad se basa en concebirlo desde el usuario hacia el propio sistema, optimizando el desarrollo en función de lo que el usuario quiere, puede o necesita hacer, en lugar de verlo desde el sistema hacia fuera, forzando a los usuarios a cambiar sus hábitos o adaptarlos a su funcionamiento.

## 2.2 El Diseño Centrado en el Usuario.

El enfoque DCU ha cobrado especial relevancia en los últimos años. No obstante, ya en 1986, Donald A. Norman y Stephen W. Draper publicaron el libro *User Centered System Design; New Perspectives on Human-Computer Interaction* en el que se trata el diseño de sistemas informáticos desde el punto de vista de los usuarios (Norman & W. Draper , 1986).

En este libro no se define un proceso como tal, pero si que presenta algunos principios generales para obtener sistemas usables, como por ejemplo, mostrar al usuario qué acciones se pueden hacer en cada estado o revelar al usuario el estado del sistema en todo momento.

El ISO 13407 provee una guía de actividades para el desarrollo de sistemas interactivos. Persigue el objetivo de crear sistemas (tanto software, como hardware) teniendo en cuenta las necesidades de los usuarios.

El estándar identifica cuatro principios:

1. La adecuada identificación de las tareas que el usuario va a realizar con el sistema basada en la participación activa de los usuarios y la comprensión de las tareas que ellos van a realizar.
2. Participación activa de los usuarios en el desarrollo del sistema.
3. El diseño de soluciones de forma iterativa, incluyendo el *feedback* de los usuarios en cada nueva iteración.



4. Equipos de diseño multidisciplinarios. Inclusión de diferentes roles en el proceso de creación del sistema.

El enfoque DCU divide el ciclo de vida en cuatro actividades que deben realizarse de forma iterativa (ver Fig. 12):

1. Comprender y especificar el contexto de uso.
2. Especificar los requisitos del usuario y de la organización.
3. Producir el diseño de varias soluciones.
4. Evaluar las soluciones contra los requisitos.

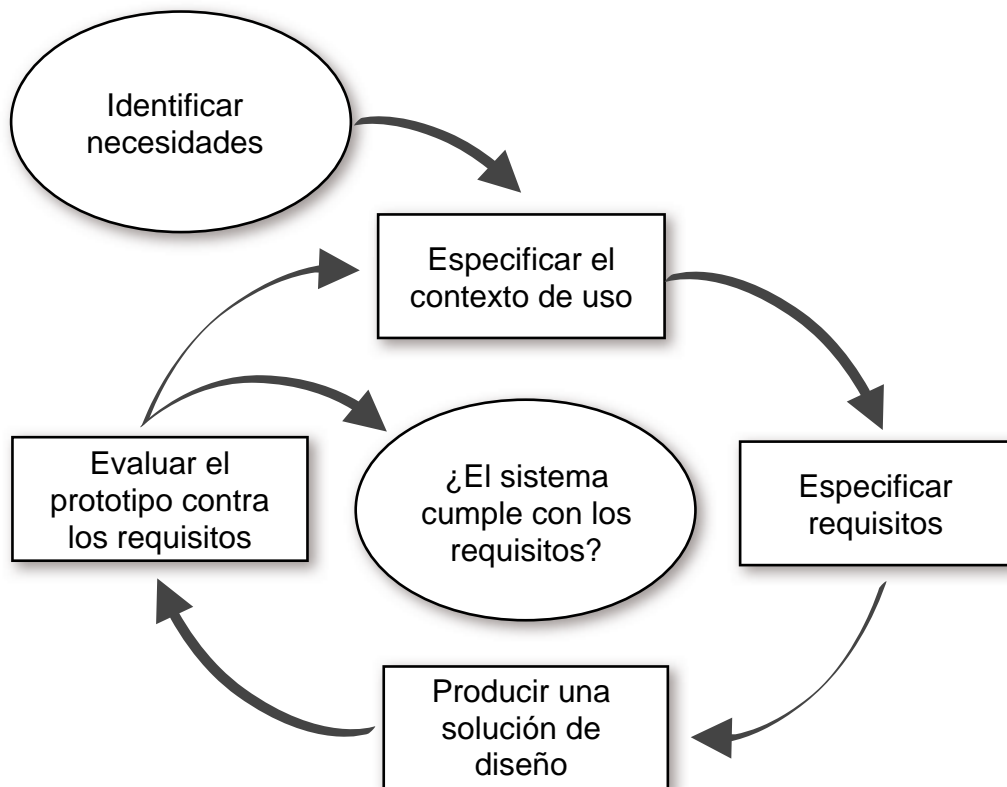


Fig. 12. Actividades del proceso de Diseño Centrado en el Usuario (ISO 13407).

La creación de sistemas, mediante esta metodología persigue:

- Motivar e incrementar la productividad.
- Mejorar la calidad del trabajo.
- Reducir los costes derivados del soporte.
- Mejorar la salud y la seguridad de los usuarios.

La meta de esta metodología es asegurar que en la conceptualización, análisis de los requisitos, diseño, evaluación y uso de un sistema se han tenido en cuenta las necesidades de los usuarios. Permite a los desarrolladores y, en general, a las partes interesadas analizar el comportamiento del sistema desde el punto de vista del usuario, teniendo en cuenta el contexto de uso.

El proceso es iterativo hasta que el sistema cumple todos los requisitos y se llega al nivel de usabilidad esperado. El comienzo del proceso consiste en identificar y planificar las actividades, técnicas y métodos que son necesarios incorporar en el proyecto. Conocer todas las técnicas y métodos de usabilidad, saber identificar cual de ellas son las más apropiadas y, en un contexto organizacional, estimar los costes que requiere su aplicación, no es una tarea sencilla.

### 2.2.1 Elementos del modelo.

El ISO/TR 18529 provee un modelo para la aplicación del enfoque DCU en un contexto organizacional. Ha surgido de la necesidad de mejorar la forma en la que se aplica el enfoque DCU descrita en el ISO 13407.

El ISO/TR 18529 describe un conjunto de procesos y subprocesos necesarios para crear sistemas usables, en concreto, identifica 7 procesos (ver Fig. 13). Cada uno de ellos representa qué se debe hacer para representar e incluir a los usuarios durante el ciclo de vida.

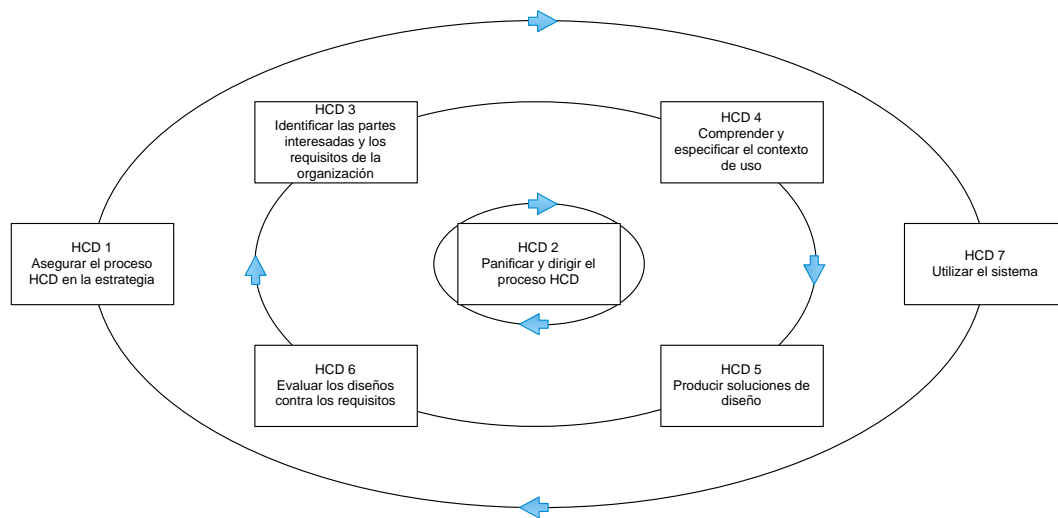


Fig. 13. Procesos identificados por el ISO/TR 18529 en el ciclo de vida.

El gráfico de la Fig. 13 representa la naturaleza iterativa de los procesos DCU y como se enlazan unos con otros.

Lo primero que cabe resaltar de este modelo es que los procesos no deben confundirse con etapas del ciclo de vida. Los procesos pueden estar presentes en más de una etapa del ciclo de vida. Por ejemplo los procesos DCU 3-6 son más técnicos y deben repetirse en bucle durante la etapa de desarrollo. En el centro tenemos el proceso DCU 2 que cubre la parte de gestión y dirección de las actividades del enfoque DCU, es el proceso central que vela por la correcta ejecución del enfoque DCU. Las soluciones de diseño que se obtienen de las iteraciones DCU 3-6 se deben contrastar con los límites y las metas que se persiguen en el proyecto, definidas en el DCU 1. El objetivo del proceso DCU 7 es probar y usar el sistema.

Cada uno de los procesos presentados se dividen en prácticas. Las prácticas son subprocesos que describen las necesidades que se deben tener en cuenta para desarrollar correctamente el proceso. Estas pueden ser métodos, técnicas o herramientas.

A continuación, se recoge el propósito y las prácticas que se definen en cada proceso:

**- DCU 1: Asegurar el enfoque DCU en la estrategia:**

- **Propósito:** Establecer y mantener el foco en los usuarios en cada parte del proceso. Como resultado de una buena implementación de este proceso:
  - La usabilidad, la ergonomía y aspectos socio-técnicos estarán presentes en el plan de marketing.
  - Las necesidades y expectativas de los usuarios estarán recogidas en el sistema.
  - Los sistemas estarán mas preparados para los cambios.
  - Los sistemas serán mejor acogidos en los mercados.
- **Prácticas:**
  - DCU 1.1. Representar a los usuarios a los que va dirigido el sistema.
  - DCU 1.2. Investigar cuales son las necesidades de los usuarios a los que va dirigido el sistema.
  - DCU 1.3. Definir una estrategia para la implementación del sistema.
  - DCU 1.4. Optimizar y refinar la estrategia del sistema en base a la opinión de los usuarios.
  - DCU 1.5. Analizar tendencias y cambios en los usuarios, en las tareas y en el contexto. Estimar necesidades futuras.

**- DCU 2: Planificar y dirigir el proceso DCU.**

- **Propósito:** Especificar cómo las actividades del enfoque DCU encajan en el ciclo de vida del proyecto. Como resultado de una buena implementación de este proceso:

- La opinión de los usuarios estará recogida en cada iteración del proyecto.
- Los recursos se emplearán para una comunicación efectiva en el equipo de diseño.
- Las técnicas y métodos de usabilidad serán apoyadas y promovidas dentro de la organización.

- **Prácticas:**

- DCU 2.1. Involucrar y hacer partícipes a los usuarios y al resto de partes interesadas.
- DCU 2.2. Planificar e identificar la mejor forma para involucrar a los usuarios en el proceso.
- DCU 2.3. Seleccionar técnicas y métodos de usabilidad que maximicen los beneficios y minimicen los riesgos para el proyecto<sup>1</sup>.
- DCU 2.4. Establecer equipos multidisciplinarios y asegurar el enfoque DCU en el equipo.
- DCU 2.5. Planificar actividades, técnicas y métodos que se deben integrar en el ciclo de vida del proyecto.
- DCU 2.6. Dirigir las actividades del enfoque DCU.
- DCU 2.7. Promover el enfoque DCU dentro de la compañía.
- DCU 2.8. Facilitar soporte para el desempeño del enfoque DCU.

**- DCU 3: Identificar las partes interesadas y los requisitos de la organización**

- **Propósito:** Establecer los requisitos de la organización y de las partes implicadas en el sistema. Como resultado de una correcta implementación de este proceso:

---

<sup>1</sup> Usability Planner daría soporte a esta práctica.

- La comunicación entre los usuarios y otras partes implicadas estará establecida.
- Los puestos de trabajo (incluyendo el reparto de tareas, la motivación, seguridad y el confort) quedarán definidos.
- La estructura y las prácticas de la organización estará definida.
- La factibilidad del proceso y su mantenimiento estará establecido.
- **Prácticas:**
  - DCU 3.1. Identificar y describir los objetivos que los usuarios se proponen alcanzar al utilizar el sistema.
  - DCU 3.2. Analizar e identificar los roles de cada grupo de usuarios y cómo afectan en el sistema.
  - DCU 3.3. Revisar la seguridad y riesgos a los que se pueden enfrentar los usuarios en el sistema.
  - DCU 3.4. Definir el uso que se le va a dar al sistema.
  - DCU 3.5. Definir los requisitos del sistema y de la organización.
  - DCU 3.6. Definir un criterio medible para asegurar la calidad del sistema.

**- DCU 4: Comprender y especificar el contexto de uso.**

- **Propósito:** Identificar, clarificar y recoger las características del entorno (físico u organizacional) en el que el sistema va a funcionar. Como resultado de una correcta implementación de este proceso:
  - Las características de los usuarios quedarán definidas.
  - Las tareas que los usuarios van a realizar quedarán definidas.
  - El contexto de uso (físico u organizacional) quedará definido.

- **Prácticas:**

- DCU 4.1. Documentar las tareas que el usuario va a realizar en el sistema.
- DCU 4.2. Documentar las características de los usuarios que van hacer uso del sistema.
- DCU 4.3. Describir y documentar el entorno organizacional.
- DCU 4.4. Documentar los dispositivos con los que va a interactuar el usuario.
- DCU 4.5. Documentar el medio físico en el que el usuario va interactuar con el sistema.

**- DCU 5: Producir soluciones de diseño.**

- **Propósito:** Crear prototipos en base a toda la información obtenida de los anteriores procesos. Como resultado de una correcta implementación de este proceso:
  - El diseño contemplará todos los requisitos identificados, tanto técnicos como sociales.
  - Las características de los usuarios y sus necesidades estarán recogidas en el diseño.
  - La comunicación entre las partes interesadas en el sistema será más fluida debido a que las decisiones de diseño son más explícitas.
  - La opinión de los usuarios y otras partes interesadas estará recogida en el diseño.
  - Se habrán creado múltiples diseños y alternativas.
  - La interfaz del sistema se habrá definido.

- **Prácticas:**

- DCU 5.1. Analizar el contexto de uso y los requisitos del sistema para definir las funciones que deben realizar el equipo, el sistema y la organización.
- DCU 5.2. Producir un modelo de tareas factible.
- DCU 5.3. Definir y analizar opciones de diseños para cada parte del sistema.
- DCU 5.4. Generar el diseño en base a toda la información recopilada, junto a guías de estilo, estándares, patrones...
- DCU 5.5. Definir cómo el sistema será usado.
- DCU 5.6. Crear prototipos que simulen el sistema.
- DCU 5.7. Identificar el entrenamiento requerido para realizar con éxito las tareas en el sistema.
- DCU 5.8. Especificar el soporte que el sistema va a dar al usuario.

- **DCU 6: Evaluar los diseños contra los requisitos.**

- **Propósito:** El propósito de este proceso es recoger la opinión de los usuarios sobre el diseño obtenido del proceso anterior. Como resultado de una correcta implementación de este proceso:
  - Los comentarios de los usuarios permitirán mejorar el diseño.
  - El uso del sistema será monitorizado.
  - Problemas potenciales serán identificados.
  - Identificar cuál de las opciones de diseño propuestas encaja mejor con los requisitos de los usuarios y de la organización.



- **DCU 7: Utilizar el sistema.**

- **Propósito:** El propósito de este proceso es introducir el sistema y dar soporte para su uso. Como resultado de una correcta implementación de este proceso:
  - Las necesidades de los usuarios estarán resueltas en el sistema.
  - La gestión de los cambios estará especificada.
  - El soporte hacia los usuarios estará establecido.
  - El soporte para la configuración del sistema en el entorno del usuario estará disponible.
  - Las reacciones y opiniones de los usuarios serán recogidas para futuras versiones del sistema.
- **Prácticas:**
  - DCU 7.1. Especificar y dirigir la gestión del cambio (reorganización del equipo, formación, reparto de responsabilidades).
  - DCU 7.2. Determinar el impacto del sistema dentro la organización y en el contexto de los usuarios.
  - DCU 7.3. Proveer soporte para la adaptación y configuración del sistema en el contexto del usuario.
  - DCU 7.4. Proporcionar soporte y formación a los usuarios y facilitar la transición al nuevo sistema.
  - DCU 7.5. Mantener el contacto con los usuarios y el cliente.
  - DCU 7.6. Comprobar que el resultado está de acorde a la legislación vigente.

## 2.3 Técnicas de usabilidad para el soporte del proceso DCU.

Para llevar cada uno de los procesos y subprocesos descritos anteriormente es necesario el uso de técnicas y métodos de usabilidad. Existen numerosas clasificaciones, algunas de ellas bastante conocidas, como la clasificación por tipo de actividades de ingeniería del software o por momentos de desarrollo en el Marco de Integración de la Usabilidad en el Proceso de Desarrollo Software (Ferré, 2005), por actividades DCU o bien por sus propiedades intrínsecas, tal y como se especifica en el ISO/TR 16982.

A continuación se detalla la clasificación ofrecida en el ISO/TR 16982 por ser la utilizada, como punto de partida, en Usability Planner.

### 2.3.1 Clasificación del ISO/TR 16982.

Este informe proporciona una categorización de técnicas y métodos de usabilidad basada en sus propiedades o características intrínsecas. Lo más destacable es que establece el escenario en el que es recomendable aplicar cada una de las técnicas.

Cada método es descrito por sus fortalezas y debilidades además de por otros factores relevantes a su selección y puesta en práctica. Además contempla las implicaciones que tiene el uso de cada método en cada una de las etapas del ciclo de vida.

El criterio propuesto se basa, principalmente, en el conocimiento de las técnicas y sus características. Así por ejemplo, la primera distinción divide al conjunto de métodos en dos grupos (ver Tabla 1):

- a) Métodos que implican la participación directa del usuario.
- b) Métodos que no requieren la participación directa de los usuarios.

Además, la clasificación se basa en los siguientes parámetros (ver Tabla 2):

- a) Etapas del ciclo de vida.
- b) Características del proyecto.
- c) Características de los usuarios.

- d) Características de las tareas que los usuarios van a realizar.
- e) Las características del sistema o producto.
- f) El grado de experiencia en usabilidad dentro del equipo.

El grado de ajuste de un método a los parámetros anteriores está cuantificado según las siguientes categorías:

- Recomendado.
- Apropiado.
- Neutro.
- No recomendado.
- No aplicable.

La siguiente tabla establece la clasificación de grupos o categorías de métodos, en función de la participación activa de los usuarios, junto a una breve descripción.

Tabla 1. Clasificación de métodos según la implicación de los usuarios.

<b>Nombre del método (o familia de métodos)</b>	<b>Implicación directa de los usuarios</b>	<b>Breve descripción del método</b>
Observación de los usuarios	Sí	Recopilación sistemática de información relativa al comportamiento y desempeño de los usuarios en la realización de una serie de tareas en un determinado contexto de uso.
Análisis y medida de tareas	Sí	Recopilación de medidas del comportamiento de los usuarios en la realización de tareas en términos de eficiencia y eficacia.
Análisis de incidentes críticos	Sí	Recopilación sistemática y posterior análisis de eventos (positivos o negativos) sobre el uso del sistema.
Cuestionarios	Sí	Recopilación de la opinión de los usuarios acerca de la interfaz de usuario mediante cuestionarios predefinidos.
Entrevistas	Sí	Método similar a los cuestionarios pero la opinión de los usuarios es recogida mediante una entrevista personal.
Pensamiento en voz alta	Sí	Consiste en la recopilación de las ideas, creencias, expectativas, dudas... que el usuario va verbalizando continuamente en voz alta durante un test de usuario.
Diseño y evaluación colaborativa	Sí	Métodos que implican a diferentes tipos de participantes (usuarios, desarrolladores y expertos en usabilidad) con el fin de realizar una evaluación o diseño del sistema.
Métodos creativos	Sí/No	Se utilizan principalmente para crear nuevas características del sistema (generalmente son los usuarios quienes aportan nuevas ideas).
Evaluación heurística	No	Evaluación dirigida por el experto en usabilidad en base a hojas de comprobación con el objetivo de dar una valoración profesional del sistema.
Métodos basados en prototipos	No	Realización de representaciones abstractas del sistema para predecir la respuesta hacia ellos de los usuarios.
Evaluación experta	No	Evaluación basada en el conocimiento y experiencia del especialista en usabilidad.
Evaluación automática	No	Evaluación mediante algoritmos basados en criterios de usabilidad para el diagnóstico y detección de deficiencias en base a unas determinadas reglas predefinidas.

Tabla 2. Categorización propuesta de los métodos en el ISO/TR 16982.

Restricciones	Categorización de métodos											
	Observación de usuarios	Análisis y medi- da de tareas	Análisis de inci- dentes críticos	Cuestionarios	Entrevistas	Pensamiento en voz alta	Diseño y evalua- ción colaborativa	Métodos de crea- tividad	Evaluación heu- rística	Métodos basados en prototipos	Evaluación ex- perta	Evaluación au- tomática
<b>1. Etapa del ciclo de vida</b>												
Conceptualización y análisis	++	+	+	+	+		+		++		+	
Desarrollo – análisis de requisitos	++	+	+	++	++	++	+	+	+	+	+	
Desarrollo – diseño	+	++		+	+	++	+	++	++	+	+	+
Desarrollo – pruebas	+	++	+	++	++	+	+		+	+	+	+
Mantenimiento – puesta en producción	+	+	+	+	+		+				+	
<b>2. Características y entorno del proyecto</b>												
Tiempo muy ajustado		–	–	–		–	–		+	–	++	+
Presupuesto ajustado		–	–		–	–		–	++	–	+	
Se requiere un alto nivel de calidad	++	++	+	++	++	+	+	+	+	+	+	+
Se necesita desde el comienzo informa- ción y diagnóstico	+			+	++		+	+			+	
Especificaciones requieren de una alta involucración	+	+	+	+	+	+	++	+				
<b>3. Características de los usuarios</b>												
No es posible su participación	NA	NA	NA	NA	NA	NA	NA	NA	+	+	+	+
Es posible su participación	++	++	+	++	++	+	++	+	+	+	+	+
Tienen una discapacidad significativa	++	+	+	+	++	+	++	+	+	–	+	–
<b>4. Características de las tareas que va a resolver el sistema</b>												
Las tareas son altamente complejas	+	+	++	+	++	++	+	+		+		
Los errores conllevan consecuencias graves	++	++	++	+	+	+	+		+	++	+	
Tareas nuevas para los usuarios	+		NA				++	++	+	+	+	
El espectro de tareas es muy amplio	+	+	+	++	+	+	+	+	++	+	+	++
Hay cambios importantes organiza- cionales o técnicos	+	+	+	+	+	+	++	++	+	–	+	–
Se requiere de altos niveles de tiempo y precisión para la interacción	+	++	++			–	–	–	–	+	–	–
<b>5. Características del producto o sistema</b>												
Adaptación de un producto existente	+	++	++	++	+	+	+		++	++	+	+
Es un producto bien comprendido	+	+		++	+		+		++		++	
Es un producto altamente personalizable	+	+	+	+	++	+	++	+				
<b>6. Habilidades y experiencia del equipo de desarrollo</b>												
Los diseñadores tienen una amplia experiencia en ergonomía y usabilidad	++	++	++	++	++	++	++	++	++	++	++	++
La experiencia en ergonomía y usabili- dad de los diseñadores es limitada	+	–	–	+	–	–	+	+	+	–	NA	+

**Leyenda:** (++) Recomendado, (+) Apropiado, (vacío) Neutral, (–) No recomendado, (NA) No aplicable.

La Tabla 2 recoge el criterio que se utilizó como punto de partida para el motor de inferencia de Usability Planner para la sección de métodos de usabilidad en función de la etapa del ciclo de vida y de las restricciones relacionadas con el entorno del proyecto, los usuarios, las tareas, las características del producto y las habilidades del equipo de desarrollo.

Su lectura es de la siguiente forma:

1. Primero, se seleccionan los métodos y técnicas de usabilidad recomendados en función de la etapa del ciclo de vida y de las características o restricciones que afectan al proyecto (grupos 1 y 2 de la Tabla 2).
2. Refinar la selección anterior en base al resto de restricciones:
  - a. Características de los usuarios (grupo 3 de la Tabla 2).
  - b. Tipo de tareas que el sistema realiza (grupo 4 de la Tabla 2).
  - c. Características del producto o sistema (grupo 5 de la Tabla 2).
  - d. Habilidades del equipo (grupo 6 de la Tabla 2).

Por ejemplo, si se necesitasen conocer los tipos de técnicas o métodos que son recomendados en la fase de conceptualización para un proyecto en el que es viable el acceso a usuarios tendríamos como recomendados (++) y apropiados (+):

- Observación de usuarios (++).
- Evaluación heurística (++).
- Análisis y medidas de tareas (+).
- Análisis de incidentes críticos (+).
- Cuestionarios (+).
- Entrevistas (+).
- Diseño y evaluación colaborativa (+).
- Evaluación experta (+).

Si el proyecto destaca por tener un presupuesto muy ajustado, el grupo de métodos de observación de usuarios quedaría descartado, permaneciendo la evaluación heurística como método recomendado (pero en menor medida).

De los métodos apropiados sólo quedarían los cuestionarios, diseño y evaluación colaborativa y la evaluación experta.

Si el equipo de desarrollo del sistema tiene poca experiencia en usabilidad y ergonomía, la evaluación heurística sigue siendo el método que de todos es más recomendado, pero nuevamente en menor medida, junto con los cuestionarios, el diseño y evaluación colaborativa. La evaluación experta quedaría descartada.

El criterio de selección de técnicas y métodos de usabilidad propuesto por el ISO tiene los siguientes inconvenientes:

- Se basa únicamente en la fortaleza o debilidad de cada método, no teniendo en cuenta otros criterios como el impacto o coste que conlleva su aplicación en un ámbito organizacional.
- El informe técnico presenta la categorización por grupos de métodos, no por técnicas y métodos en particular.
- Se necesita bastante experiencia en usabilidad para hacer una correcta interpretación del criterio de selección.

### 2.3.2 Clasificación del ISO/WD 9241-230.

Este informe (actual borrador del ISO/TR 16982) realiza la categorización de métodos en función de si existe acceso a los usuarios, si se trata de un sistema en producción o bien es un prototipo u otra representación, si es un producto o servicio y si existe interés en su análisis, diseño y evaluación.

Esta aproximación provee una categorización más cercana al mundo profesional (ver Tabla 3).

Tabla 3. Tipos de métodos del ISO/WD 9241-230.

	Métodos que implican la participación directa de los usuarios.	Métodos que no implican la participación directa de los usuarios.
<b>Métodos que requieren un sistema</b>	<ul style="list-style-type: none"> <li>• Observación del uso real.</li> <li>• Test de usuario.</li> </ul>	<ul style="list-style-type: none"> <li>• Evaluación basada en la inspección.</li> <li>• Evaluación automática.</li> <li>• Análisis de datos de uso</li> <li>• Diseño de la interfaz y de la interacción.</li> </ul>
<b>Métodos que pueden ser aplicados sin necesidad de un sistema</b>	<ul style="list-style-type: none"> <li>• Observación de los usuarios en su contexto.</li> <li>• Entrevistas.</li> <li>• Cuestionarios.</li> <li>• Métodos colaborativos.</li> <li>• Reporte propio de los usuarios.</li> </ul>	<ul style="list-style-type: none"> <li>• Evaluar la información relacionada con el sistema.</li> <li>• Enfoque basada en modelos.</li> <li>• Métodos analíticos.</li> </ul>

### 2.3.3 Clasificación por etapas del ciclo de vida.

Las técnicas y métodos pueden ser clasificados por etapas del ciclo de vida, siendo una aproximación más cercana al desarrollo de sistemas. Aunque tiene un inconveniente, el hecho de que una técnica o método pueda repetirse en distintas etapas del ciclo de vida, es la aproximación más natural en un contexto profesional o empresarial.

En las dos clasificaciones vistas anteriormente se habla de tipos de métodos en lugar de métodos y técnicas concretas. En esta propuesta se puede llegar a un mayor nivel de detalle ya que un método puede estar compuesto por sub-métodos. Por ejemplo, el test de usabilidad puede estar compuesto por la observación de los usuarios, medidas de rendimiento, cuestionarios, entrevistas y pensamiento en voz alta.

El criterio sugerido y que implementa Usability Planner comienza por identificar las actividades o etapas del ciclo de vida para, seguidamente, seleccionar los métodos más apropiados en función del perfil de usuario que utilice la aplicación:



1. Primero, identificar las buenas prácticas que, en términos de usabilidad, son viables para incrementar los beneficios y minimizar los riesgos del proyecto. Para ello se puede hacer uso de las buenas prácticas recogidas en el ISO PAS 18152.
2. Segundo, identificar los métodos y técnicas que son más recomendadas para conseguir las buenas prácticas anteriormente identificadas:
  - a. En qué medida cada método contribuye a cumplir cada buena práctica.
  - b. Cómo de rentable es la aplicación de cada método en términos de:
    - i. Restricciones como el tiempo, el presupuesto, acceso a las partes interesadas del proyecto, a los usuarios...
    - ii. La naturaleza de las tareas que el sistema resuelve, complejidad, formación requerida, consecuencia de los errores...
    - iii. La naturaleza del producto o sistema, cuánto de complejo es, si es nuevo o es una revisión de un sistema actual...
    - iv. El contexto de uso o rango de contextos.

Para completar este paso Bevan especifica que debe realizarse en base a la experiencia del consultor en usabilidad o con ayuda de documentación (Bevan, 2009). En este punto es dónde surge la idea de crear Usability Planner, para disminuir esta complejidad y para que personas con menor experiencia en usabilidad puedan elaborar su plan sin necesidad de consultar otras fuentes.

En el siguiente gráfico (ver Fig. 14) se resume este proceso.

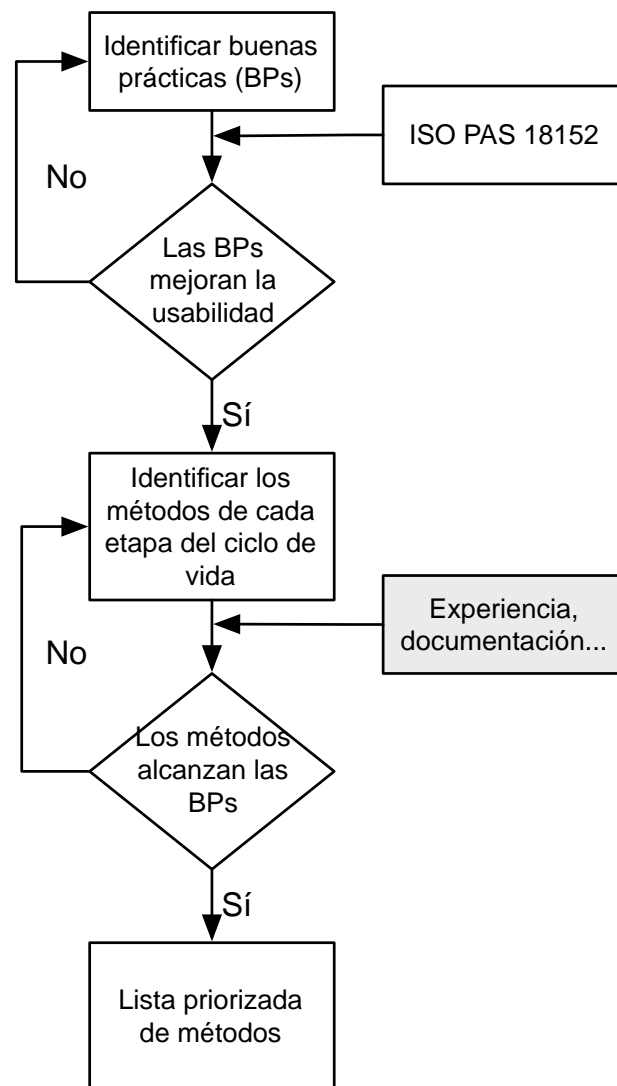


Fig. 14. Algoritmo del criterio para la selección de métodos.

En el primer anexo de este documento se incluyen ejemplos de técnicas de usabilidad que pueden ser usadas para dar soporte a las buenas prácticas, para minimizar riesgos en el desarrollo de sistemas.

A continuación (ver Tabla 4) se incluye un extracto de la tabla propuesta por Bevan y Ferré con la priorización de las técnicas en base a la etapas del ciclo de vida y en función de las limitaciones que afectan al proyecto, los usuarios, las tareas, el producto, el contexto y la experiencia del equipo.

Tabla 4. Extracto de la clasificación por etapas del ciclo de vida propuesta por Bevan y Ferré para perfiles profesionales ligados a la usabilidad.

Etapas del ciclo de vida	Ejemplo de técnicas y métodos	Restricciones del proyecto				Restricciones de los usuarios				(...)
		Se necesitan resultados rápidos	Presupuesto muy ajustado	La usabilidad es clave	Especificaciones poco claras	Difícil participación de usuarios	No hay acceso a usuarios	Usuarios discapacitados	Usuarios primarios	(...)
1. Conceptualización	- Reuniones de trabajo para identificar necesidades futuras.	2	2	4	5	2	1	3	5	(...)
1.1. Prever oportunidades.	- Estudios de campo preliminares.	4	4	5	5	2	1	4	3	
	- Grupos de enfoque.	3	2	5	5	2	1	3	5	
	- Estudios con fotografías.	4	4	5	5	2	1	4	3	
	- Simulaciones del contexto o entorno futuro.	2	2	5	5	2	1	3	3	
	- Tormenta de ideas.	5	5	5	5	3	3	3	3	
	- Tormenta de ideas (mediante dibujos).	5	5	5	5	3	3	3	3	
	- Análisis en profundidad del trabajo y estilo de vida de los usuarios.	1	1	5	5	1	1	2	3	
1.2 Alcance del sistema	- Reuniones de trabajo participativas.	4	3	5	5	2	1	4	4	(...)
	- Estudios de campo o etnográficos.	1	1	5	5	2	1	4	3	
	- Consulta a las partes interesadas.	4	4	5	5	5	5	5	4	
	- Análisis de factores humanos.	2	2	5	3	4	4	3	3	
	- Análisis del contexto de uso.	3	5	5	5	3	3	3	3	
2. Planificación	- Plan para alcanzar y mantener la usabilidad.	3	3	5	3	3	3	3	3	(...)
3. Captura de requisitos	Identificación de partes interesadas.	5	5	5	3	5	5	3	3	(...)
3.1 Contexto de uso	- Estudios de campo o etnográficos.	1	1	5	5	2	1	4	3	
	- Reuniones de trabajo participativas.	2	2	5	5	2	1	4	4	
	- Análisis del contexto de uso.	3	5	5	5	3	3	3	3	
	- Análisis del contexto de trabajo.	3	5	5	5	3	3	3	3	
	- Análisis de datos por eventos.	1	4	4	3	5	5	2	3	
	- Definición del contexto de uso y su alcance.	1	1	5	5	2	1	3	4	
	- Diarios.	1	2	4	3	2	1	3	1	
(...)	(...)	(...)				(...)				(...)

**Leyenda:** (5) Recomendado, (4) Apropiado, (3) Neutral, (2) Poco apropiado, (1) Inapropiado.

La Tabla 4 es un extracto de la clasificación o criterio para la selección de técnicas o métodos y la base del motor de inferencia de Usability Planner para elaborar el ranking o priorización de métodos. Sólo se han incluido las tres primeras etapas del ciclo de vida y dos limitaciones, las asociadas al proyecto y a los usuarios. La lista completa de etapas del ciclo de vida se detalla en el segundo anexo de este documento.

La lista completa de restricciones que utiliza Usability Planner es:

- a) Restricciones asociadas al proyecto.
- b) Restricciones asociadas a los usuarios.
- c) Restricciones asociadas a las tareas que los usuarios van a realizar.
- d) Restricciones asociadas al sistema o producto
- e) Restricciones asociadas al grado de experiencia en usabilidad dentro del equipo de desarrollo.

Estas restricciones coinciden con las mencionadas en el ISO TR 16982. El grado en el que una técnica o método se adecua a una etapa del ciclo de vida en función de las restricciones anteriores está cuantificado de la siguiente forma:

- 5 puntos: Recomendado.
- 4 puntos: Apropiado.
- 3 puntos: Neutral.
- 2 puntos: Poco apropiado
- 1 punto: Inapropiado.

Por ejemplo, si se necesitasen conocer las técnicas o métodos que se recomiendan en la etapa de **conceptualización** para un proyecto en el que es viable el acceso a usuarios tendríamos como recomendados:

- Reuniones de trabajo para identificar reuniones futuras.
- Estudios de campo preliminares.
- Grupos de enfoque.

- Estudio con fotografías.
- Simulaciones del entorno o contexto futuro.
- Tormenta de ideas mediante dibujos.
- Tormenta de ideas.
- Análisis en profundidad del trabajo y del estilo de vida de los usuarios.
- Estudios de campo.
- Reuniones de trabajo participativas.
- Consulta a las partes interesadas.
- Análisis de factores humanos.
- Análisis del contexto de uso.

Si además el proyecto destaca por tener un presupuesto muy ajustado, de las técnicas y métodos anteriores serían viables las siguientes:

- Tormenta de ideas con dibujos.
- Tormenta de ideas.
- Análisis del contexto de uso.
- Consultar a las partes interesadas.
- Reuniones de trabajo participativas.

Si el equipo de desarrollo del sistema tiene poca experiencia en usabilidad y ergonomía la tormenta de ideas con dibujos perdería peso, aunque sería viable.

El criterio recogido en la Tabla 5 se basa en las etapas del ciclo de vida acordes a un perfil de usuario relacionado directamente con el ámbito de la usabilidad (investigadores, profesionales y estudiantes). Usability Planner además da soporte a la priorización de técnicas y métodos para perfiles más ligados al ámbito del desarrollo. Para este caso se han empleado las etapas del ciclo de vida del Proceso Unificado de Rational (ver segundo anexo).

Tabla 5. Extracto del criterio de selección para perfiles de desarrollo.

Etapas del ciclo de vida	Ejemplo de técnicas y métodos	Restricciones del proyecto				Restricciones de los usuarios				(...)
		Se necesitan resultados rápidos	Presupuesto muy ajustado	La usabilidad es clave	Especificaciones poco claras	Difícil participación de usuarios	No hay acceso a usuarios	Usuarios discapacitados	Usuarios primerizos	(...)
1. Inicio	- Reuniones de trabajo para identificar necesidades futuras.	2	2	4	5	2	1	3	5	(...)
	- Estudios de campo.	4	3	5	5	2	1	4	3	
	- Grupos de enfoque.	3	2	5	5	2	1	3	5	
	- Estudios con fotografías.	4	4	5	5	2	1	4	3	
	- Simulaciones del contexto o entorno futuro.	2	2	5	5	2	1	3	3	
	- Tormenta de ideas.	5	5	5	5	3	3	3	3	
	- Análisis en profundidad del trabajo y estilo de vida de los usuarios.	1	1	5	5	1	1	2	3	
	- Reuniones de trabajo participativas.	4	3	5	5	2	1	4	4	
	- Estudios de campo o etnográficos.	1	1	5	5	2	1	4	3	
	- Consulta a las partes interesadas.	4	4	5	5	5	5	5	4	
	- Análisis de factores humanos.	2	2	5	3	4	4	3	3	
	- Análisis del contexto de uso.	3	5	5	5	3	3	3	3	
	- Guiones gráficos (viñetas)	2	2	4	5	4	4	3	3	
	- Identificación de partes interesadas.	5	5	5	3	5	5	3	3	
	- Análisis de la competencia.	3	2	5	5	3	3	3	3	
	- Diseño paralelo	2	2	5	5	3	3	3	3	
	- Bocetos	5	5	5	5	3	2	3	4	
	- Desarrollo de prototipos	5	5	5	5	2	1	3	4	
(...)	(...)	(...)				(...)				(...)
Leyenda: (5) Recomendado, (4) Apropiado, (3) Neutral, (2) Poco apropiado, (1) Inapropiado.										

El proceso para realizar la priorización de las técnicas es similar. Por ejemplo, si un desarrollador deseara conocer qué técnicas son las más recomendadas en la etapa de inicio cuando se necesitan resultados rápidos y el presupuesto es muy ajustado, el resultado sería: Tormenta de ideas, identificación de las partes interesadas, desarrollo de prototipos, bocetos y en menor medida estudio con fotografías.

### 2.3.4 ¿En qué etapa del ciclo de vida se debe elaborar el plan de usabilidad?

En el apartado anterior se han descrito diferentes clasificaciones de técnicas y métodos de usabilidad. Todas ellas tienen un objetivo, introducir mejoras en la usabilidad de un producto.

¿En qué momento se debe diseñar un plan que priorice las técnicas y métodos que introduzcan dichas mejoras?

Ya sea la clasificación de actividades del ciclo de vida del Proceso de Diseño Centrado en el Usuario o bien la clasificación de etapas del Proceso Unificado, en la actividad o etapa de planificación se debe elaborar el plan y debe ser revisado en cada una de las actividades o etapas siguientes.

Dicho de otra forma, Usability Planner es una herramienta para hacer sencilla la etapa de planificación del proyecto, ya que de forma inmediata, en base a las características del proyecto elabora un plan.

## Capítulo 3.

# DISEÑO DE LA INTERACCIÓN.

Este capítulo describe cómo ha sido el proceso de diseño de la aplicación, desde su conceptualización hasta su puesta en producción, junto con las técnicas y métodos que se han empleado para garantizar la usabilidad.

Para su consecución, se ha seguido el proceso DCU descrito en el capítulo anterior. Uno de los pilares de esta metodología es su carácter iterativo e incremental. Cabe destacar que se han empleado cuatro iteraciones para llegar a una solución satisfactoria o a una versión estable de la aplicación. A lo largo de este capítulo se detalla cómo ha sido cada una de estas iteraciones.

Durante estas cuatro iteraciones, la aplicación ha ido evolucionando en función de los resultados ofrecidos por las pruebas de usabilidad realizadas al finalizar cada iteración. Por ejemplo, inicialmente la aplicación se pensó que fuera de escritorio, ya que el contexto de uso previsto iba a ser un entorno profesional o empresarial, pero tras evaluar el primer prototipo, se decidió reconducir el diseño hacia una aplicación web y así aprovechar las ventajas que ofrece esta plataforma.

El prototipado, la evaluación por expertos, los test de usuario, las entrevistas, personas y otras técnicas de usabilidad han sido claves para construir, a partir de una idea, una solución con un alto nivel de usabilidad.

La colaboración de usuarios de distintos perfiles y con distintas necesidades, su participación en los test de usabilidad y en las encuestas de satisfacción, junto con la opinión de expertos en el ámbito de la experiencia de usuario han sido la clave para crear una aplicación en la que el usuario ha sido el eje de toda la toma de decisiones, llegando a un resultado que responde a los objetivos marcados.



## 3.1 Primera iteración.

La idea de construir una aplicación que priorizase técnicas y métodos de usabilidad en función de las restricciones habituales que se dan en el Desarrollo Centrado en el Usuario (DCU) de un sistema surgió en septiembre de 2009, a partir del artículo escrito por Bevan (Bevan, 2009) para el congreso INTERACT de 2009.

Construir una aplicación que diera forma a los planteamientos recogidos en el artículo fue la motivación y origen de este proyecto.

### 3.1.1 Conceptualización.

Bevan pone de manifiesto en su artículo la complejidad actual para realizar una tarea: seleccionar métodos y técnicas de usabilidad en el proceso de desarrollo de software, y aunque existe una extensa bibliografía con distintos enfoques y criterios, ninguno tiene un carácter práctico en contextos organizacionales.

Como punto de partida se realizó un análisis competitivo, esto es, un estudio de aplicaciones, herramientas o *frameworks* que facilitasen esta labor. Únicamente se analizaron aplicaciones disponibles en línea. Se identificaron las siguientes:

#### **Usability Body of Knowledge<sup>2</sup>.**

Su objetivo es convertirse en un sitio de referencia que recoja el conocimiento colectivo de los profesionales que se dedican a la usabilidad. Está impulsado por la UXPA (Asociación de Profesionales dedicados a la Experiencia de Usuario).

Dispone de una base de datos de métodos (unos 40 aproximadamente) clasificados por actividades del Desarrollo Centrado en el Usuario. Cada método viene acompañado de una descripción, sus beneficios, ventajas y desventajas.

La aplicación no prioriza los métodos, el usuario debe de forma manual, elegir que métodos usar en base a la información que se detalla de cada uno de ellos. El uso que un usuario puede hacer de este portal de información es únicamente de consulta.

---

<sup>2</sup> Más información en: <http://www.usabilitybok.org>

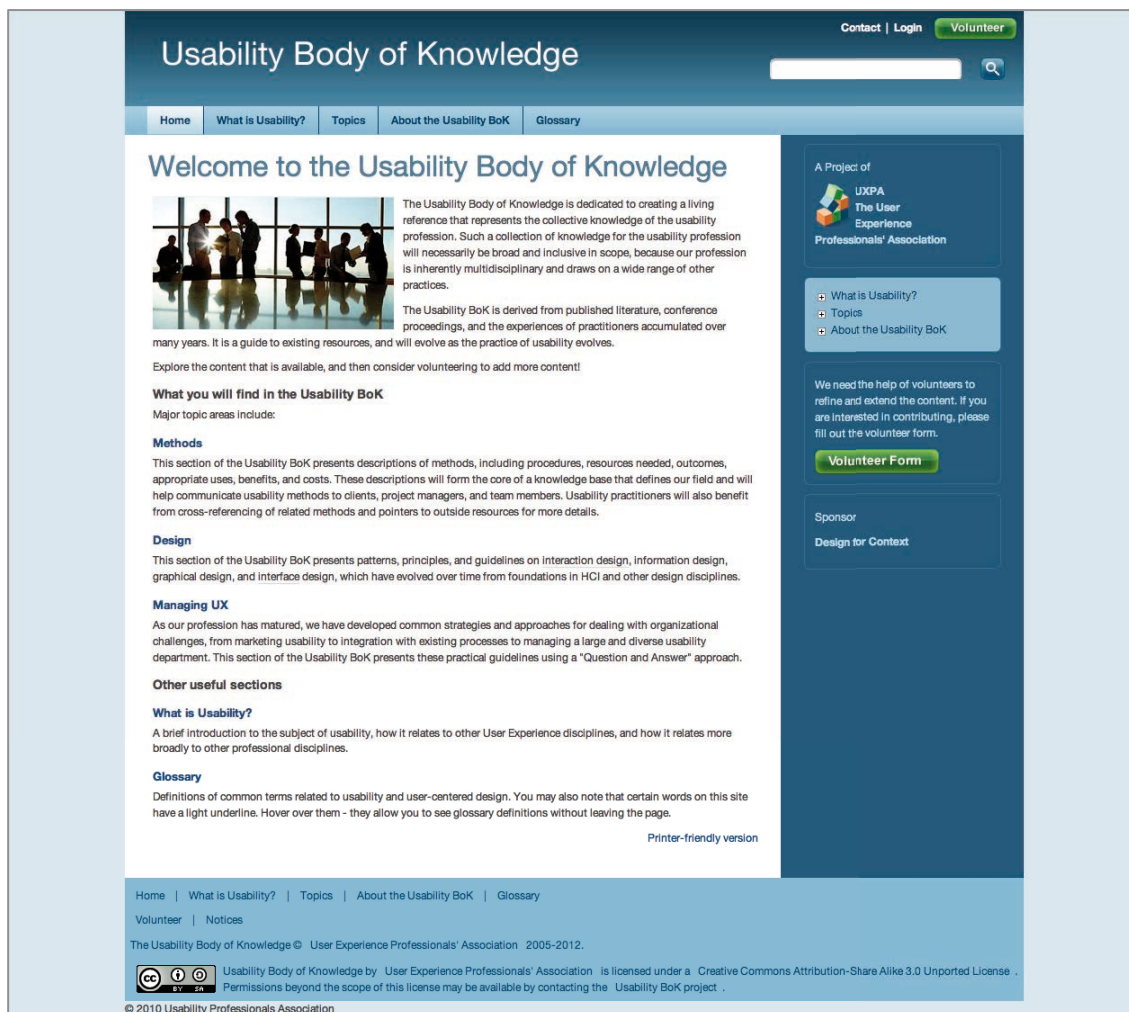


Fig. 15. Usability Body of Knowledge.

### UsabilityNet Methods Table<sup>3</sup>.

Este portal, creado por Nigel Bevan en el año 2001, provee información de utilidad para profesionales del mundo de la usabilidad, recursos y herramientas, entre ellas, una lista de métodos y técnicas clasificados por etapas del ciclo de vida. Probablemente fuera un primer intento de Bevan por crear un sistema experto que recomendase o priorizase técnicas y métodos de usabilidad.

<sup>3</sup> Más información: <http://www.usabilitynet.org/tools/methods.htm>

El listado de métodos y técnicas se puede filtrar por tres criterios:

- No hay recursos o el tiempo es limitado.
- No se tiene acceso a los usuarios.
- La experiencia o habilidad del equipo es limitada.

El proyecto está descontinuado desde el año 2006.

**Usability Net**

Home | Professional Groups | Tools & Methods | Usability Practitioner | Usability for Managers | EU Project Support | About this site

methods table | guidelines | reference material | methods list

### Methods table

you can select the most appropriate methods depending on three conditions

☐ limited time/resources ☐ No direct access to users ☐ Limited skills/expertise

Planning & Feasibility	Requirements	Design	Implementation	Test & Measure	Post Release
Getting started	User Surveys	Design guidelines	Style guides	Diagnostic evaluation	Post release testing
Stakeholder meeting	Interviews	Paper prototyping	Rapid prototyping	Performance testing	Subjective assessment
Analyse context	Contextual inquiry	Heuristic evaluation		Subjective evaluation	User surveys
ISO 13407	User Observation	Parallel design		Heuristic evaluation	Remote evaluation
Planning	Context	Storyboarding		Critical Incidence Technique	
Competitor Analysis	Focus Groups	Evaluate prototype		Pleasure	
	Brainstorming	Wizard of Oz			
	Evaluating existing systems	Interface design patterns			
	Card Sorting				
	Affinity diagramming				
	Scenarios of use				
	Task Analysis				
	Requirements meeting				

Types of methods: [Planning and feasibility](#) | [Requirements](#) | [Design](#) | [Implementation](#) | [Testing and measurement](#) | [Post release](#)

[Home](#) | [Professional Groups](#) | [Tools & Methods](#) | [Usability Practitioner](#) | [Usability for Managers](#) | [EU Project Support](#) | [About this Site](#)

©UsabilityNet 2006. Reproduction permitted provided the source is acknowledged.

Fig. 16. Tabla de métodos (Usability Net).

## Generic Work Process<sup>4</sup>.

Se trata de una tabla en el que se listan los métodos y técnicas que son aplicables en una determinada etapa del ciclo de vida. El criterio de selección es manual.

Cada técnica y método está documentado con una descripción, el propósito, puntos de interés y referencias.

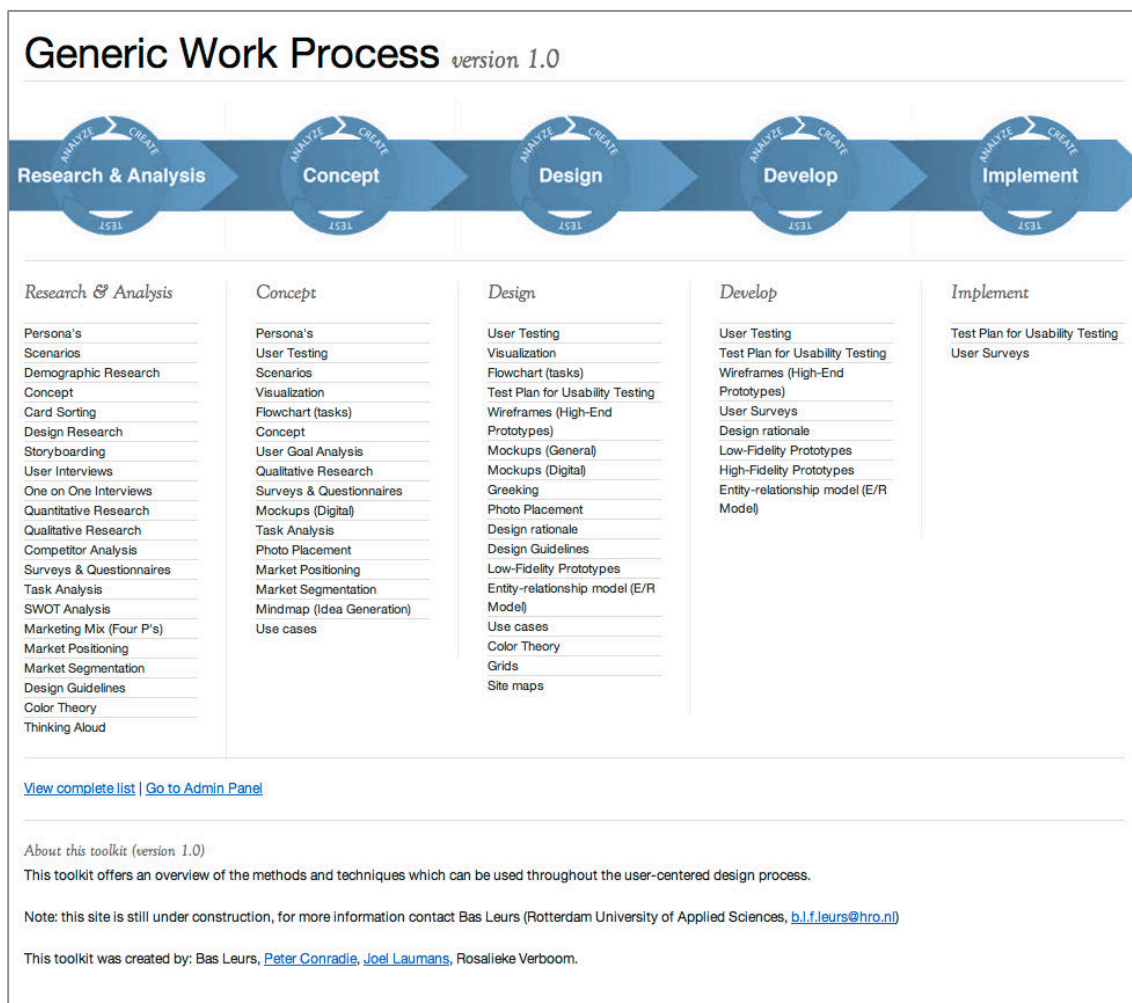


Fig. 17. Generic Work Process.

<sup>4</sup> Más información en: <http://project.cmd.hro.nl/cmi/hci/toolkit/>

## Usability.gov

Es un portal mantenido por el HSS (Dpto. de Salud y Servicios Sociales, en inglés *Department of Health and Human Services*) del gobierno de los EE.UU. dirigido a profesionales del ámbito de la usabilidad en el que se pueden encontrar numerosos recursos (plantillas, guías, artículos y discusiones).

Dispone de una clasificación de métodos basada en actividades del proceso de diseño centrado en el usuario. La clasificación se realiza por etapas del ciclo de vida, pero, al igual que en Usability Body of Knowledge, el uso que se le puede dar al portal es de consulta.



Fig. 18. Usability.gov.

Ninguna de las iniciativas anteriores propone un proceso guiado para la selección de métodos y técnicas de usabilidad, además, salvo UsabilityNet (y de forma limitada), no tienen en cuenta aspectos intrínsecos del proyecto sobre el que se van a aplicar las técnicas, como restricciones relacionadas con el acceso a usuarios, las tareas a realizar, el contexto o el tipo de proyecto.

Tampoco distinguen el perfil de usuario que va a aplicar las técnicas y ninguna prioriza los métodos en función de los costes (riesgos y beneficios) de su aplicación en un determinado proyecto.

Otro punto en común, que tienen las herramientas anteriores, es que el usuario debe tener experiencia y formación en usabilidad para poder elegir los métodos a aplicar.

### 3.1.2 Comprendiendo necesidades y capturando requisitos.

El estudio anterior reveló la necesidad de crear una aplicación que priorice las técnicas en función de un determinado proyecto. El punto de partida fue la realización de varias tormentas de ideas para extraer los requisitos que debiera tener la aplicación a crear.

Se identificaron los siguientes **requisitos funcionales** principales:

- Los usuarios podrán obtener una lista priorizada de métodos y técnicas de usabilidad (a modo de ranking) asociadas a las actividades del proceso de Desarrollo Centrado en el Usuario. Inicialmente para las actividades recogidas en el ISO PAS 18152.
- Los usuarios podrán seleccionar las etapas o sub-etapas del ciclo de vida en las que necesiten aplicar técnicas o métodos de usabilidad.
- Adicionalmente, los usuarios podrán seleccionar los beneficios o riesgos, en términos de negocio, por etapa del ciclo de vida relacionados, con su proyecto.
- Los usuarios podrán indicar las limitaciones que tiene su proyecto en función de las etapas o sub-etapas elegidas. Estas limitaciones serán relacionadas con el proyecto, con los usuarios, con las tareas, con el producto y con el contexto de uso. Se tomarán como base, inicialmente, las limitaciones incluidas en el ISO TR 16982.
- El grado de recomendación de cada técnica o método estará cuantificado.

Además de los requisitos funcionales se identificaron otros **requisitos no funcionales** o también llamados *qualities* (Stellman & Greene, 2005):

- Aplicación de escritorio. El sistema deberá poder ser ejecutado en cualquier máquina independientemente del sistema operativo. Además su ejecución no dependerá de otros sistemas o agentes externos.
- La modificación de la base de datos no requerirá ninguna experiencia en el manejo de bases de datos.
- El sistema deberá ser desarrollado con un enfoque centrado en el usuario, de forma que el resultado maximice la eficiencia, la eficacia y la comprensión, minimice el aprendizaje y genere satisfacción al usuario.
- Escalabilidad. Al ser un sistema cuyas bases son un proyecto de investigación en desarrollo, expuesto a cambios y actualizaciones, deberá facilitarse la adecuación y actualización de los contenidos de forma ágil.
- Costo. El sistema deberá ser desarrollado utilizando tecnologías de código abierto que no requieran del pago de licencias.
- Distribución. El sistema deberá ser abierto (opensource). El código fuente deberá estar a disposición de la comunidad, la cual tendría pleno derecho para desarrollar mejoras o sistemas derivados del mismo.
- Los usuarios u organizaciones podrán adaptar la aplicación a su contexto, modificando los parámetros que hacen posible la elaboración del ranking e incluyendo las técnicas que estimen necesarias.
- Idioma. El sistema deberá ser lanzado en inglés y preparado para albergar nuevos idiomas de forma sencilla.

### 3.1.3 Creación de Personas.

Para garantizar la usabilidad de un sistema se debe primero conocer a fondo a qué usuarios específicos está destinado y cuáles son sus características principales (Hix & Hartson, 1993).

Conocer cuáles son las necesidades de los usuarios y cuál es su motivación principal para utilizar el sistema son el punto de partida en todo proceso de creación software. Para llevar a cabo el análisis es necesario estar en contacto con usuarios finales, conocer su entorno, realizar entrevistas, observarles y entenderles.

Para llevar a cabo esta tarea se empleó la técnica de **Personas**. Una persona es una representación de un usuario ficticio que incluye una colección de características que representan al usuario al que va dirigida la aplicación, junto con sus experiencias, metas, tareas y su contexto de uso (Cooper, 2004). Este método permitió la identificación de los usuarios a los que se dirige el sistema.

El proceso de identificar las personas se ha realizado teniendo en cuenta los posibles contextos de uso de la herramienta, así como las tareas y necesidades que pueden tener los usuarios.

Se identificaron tres tipos de personas (o perfiles de usuario) para ilustrar los principales grupos de usuarios a los que va dirigido el sistema:

1. **Estudiantes de usabilidad** que están aprendiendo las técnicas y métodos que se aplican durante el proceso DCU. No están interesados en conocer las técnicas y métodos que maximizan beneficios y minimizan riesgos en un contexto empresarial, pero sí que pueden ver útil conocer cómo determinadas limitaciones, como el acceso a los usuarios, hacen a unas técnicas mas recomendadas que otras.
2. **Consultores junior en usabilidad** que trabajan en empresas en las que existe un departamento específico que se encarga de aplicar técnicas y métodos DCU en sus proyectos. Están interesados en conocer las técnicas de usabilidad que, en un determinado contexto, son más recomendadas en función de las restricciones que lo limitan.
3. **Consultores sénior en usabilidad** más ligados a la gestión y planificación de proyectos y preocupados por el coste económico de aplicar unas técnicas u otras. La aplicación les podría ayudar a determinar aquellas técnicas que maximicen los beneficios y minimicen los riesgos de sus proyectos. Además, en base a su experiencia, podrán ajustar la clasificación que ofrece la herramienta y configurarla en base a su contexto organizacional.

En base a estos perfiles se han creado tres personas:





Sara / 21 años / Española / Estudiante

---



*Sara es estudiante de tercero de Grado en Informática. En su Facultad, está disfrutando de una beca en Interacción Persona Ordenador en el laboratorio de Ingeniería del Software. Allí está aprendiendo las principales técnicas y métodos de usabilidad así como el papel que éstas desempeñan en el Diseño Centrado en el Usuario.*

*Sara está especialmente interesada en conocer a fondo las técnicas y métodos que se recomiendan aplicar en una determinada etapa del ciclo de vida y cómo restricciones como el tiempo o el acceso a los usuarios pueden influir en dicha selección.*

*A Sara le gustaría tener una herramienta clara y concisa que le ayudase a saber cuáles son las técnicas y métodos que mejor encajan en cada etapa del ciclo de vida. Actualmente utiliza la bibliografía disponible en su departamento y otros recursos en Internet.*

*A Sara le encanta el cine, suele ir una media de 3 veces al mes. También salir con sus amigas y bailar salsa. Dedicar un par de horas al día a navegar por Internet, para consultar su correo, ver Facebook y consultar información para realizar las prácticas de su carrera.*



Carmen / 26 años / Española / Ingeniera en Informática Superior

---



*Carmen lleva dos años trabajando en Usertic, en el departamento de experiencia de usuario. Se encarga de definir la arquitectura de información de las aplicaciones web y móviles que se desarrollan en su empresa. También se encarga de realizar prototipos de baja fidelidad de los proyectos en los que participa y analizar los resultados de los test de usabilidad que se realizan.*

*Carmen es una entusiasta de su profesión y siempre está muy interesada en todo lo que tenga que ver con la usabilidad y la experiencia del usuario. En los dos años que Carmen lleva en Usertic siempre ha aplicado las mismas técnicas en todos los proyectos, básicamente prototipado, personas, análisis de tareas y test de usabilidad.*

*Carmen piensa que no siempre estas técnicas son las más adecuadas para todos los proyectos y no encuentra una referencia clara en la que se recojan técnicas y métodos de usabilidad recomendados en función de las características del proyecto y su contexto. A Carmen le gustaría tener una aplicación de apoyo que le ayudase a identificar que técnicas aplicar y con que prioridad en los proyectos en los que participa.*

*Carmen vive con sus padres, pero tiene previsto independizarse a lo largo de este año. Le encanta la equitación, la lectura y salir los fines de semana con sus amigos.*



**Steven** / 37 años / Chino / Ingeniero Industrial

---



*Steven dirige el Departamento de Experiencias (DEX) de la empresa Usertic, cuenta con más de 10 años de experiencia en Interacción Persona-Ordenador y ha participado en proyectos de gran relevancia a nivel nacional e internacional con excelentes resultados.*

*Actualmente está al frente de un equipo de 6 consultores en experiencia de usuario y su misión es establecer y planificar las técnicas y métodos de usabilidad que cada proyecto va a requerir y organizar a su equipo para llevarlas a cabo.*

*Steven no cuenta habitualmente con mucho presupuesto para sus proyectos, por ejemplo, no puede incluir técnicas como la observación etnográfica o el test de usabilidad en laboratorio por limitaciones económicas y de tiempo.*

*A pesar de su experiencia y conocimiento le gustaría tener una aplicación que le permitiera extraer las técnicas y métodos de usabilidad que maximicen los beneficios y minimicen los riesgos en el contexto de su empresa. Una herramienta que además le permitiese adaptar los resultados en función del contexto de su organi-*

*zación y en base a su conocimiento y experiencia para agilizar su trabajo y el tiempo que actualmente emplea en tomar decisiones.*

*Se considera un adicto al trabajo y un entusiasta de su profesión. Los fines de semana dedica gran parte de su tiempo a leer artículos y libros relacionados con la experiencia de usuario. Le encanta viajar, siempre que dispone de vacaciones visita a su familia en Pekín, su ciudad natal.*



Estas son las 3 personas que se tomaron como referencia para la creación de la aplicación. Son perfiles con diferentes inquietudes y niveles de experiencia. Por este motivo, una de las características o requisitos que se establecieron fue la provisión de mecanismos flexibles para trabajar con la herramienta, independientemente del grado de experiencia o perfil profesional del usuario que interactúe con ella.

#### 3.1.4 Análisis de tareas.

En función de las personas identificadas anteriormente y de los requisitos planteados se definió una lista de tareas principales o metas a las que el sistema debía dar soporte.

A continuación, en la Tabla 6, se recogen las 5 tareas principales que se identificaron en esta iteración junto con las personas a las que van dirigidas.

Tabla 6. Tareas asociadas a las personas identificadas.

	Tarea	Sara	Carmen	Steven
1	Seleccionar actividades DCU del ISO PAS 18152 y consultar las buenas prácticas asociadas a cada etapa.	□	□	□
2	Seleccionar los beneficios o riesgos en términos de negocio que consideren en cada etapa.	×	×	□
3	Indicar las limitaciones que su proyecto tiene en función de las etapas o subetapas elegidas. Por ejemplo falta de presupuesto.	○	□	□
4	Ver priorizadas las técnicas y métodos que la herramienta recomienda.	□	□	□
5	Adaptar la aplicación en base a su contexto organizacional para afinar la lista de técnicas y métodos recomendados.	×	×	□
Leyenda: □ Tarea asociada al usuario, × Tarea no asociada al usuario, ○ Neutro.				

Como se puede ver en la tabla anterior hay tareas que están más enfocados al ámbito empresarial. Por ejemplo la tarea 2, posiblemente sólo sea del interés de Steven ya que el resto de perfiles no están ligados a aspectos de negocio.

Los pasos que los usuarios tendrían que dar para completar cada tarea son:

1. Seleccionar actividades DCU del ISO PAS 18152 y consultar las buenas prácticas asociadas a cada etapa.
  - a. El usuario puede necesitar priorizar métodos para distintas etapas del ciclo de vida. Para ello seleccionará aquellas etapas (y/o sub-etapas) del enfoque DCU.
  - b. Por cada etapa podrá consultar información sobre su momento de aplicación y el listado de buenas prácticas asociadas para garantizar la correcta aplicación del proceso.
2. Seleccionar los beneficios o riesgos en términos de negocio que consideren en cada etapa.

- a. Por cada etapa seleccionada podrá elegir aquellos riesgos y/o beneficios significativos para la correcta consecución del proyecto.
3. Indicar las limitaciones que su proyecto tiene en función de las etapas o subetapas elegidas (por ejemplo falta de presupuesto).
  - a. El sistema mostrará un listado de limitaciones o restricciones agrupadas por tipo (proyecto, usuario, tareas, producto, contexto de uso y experiencia).
  - b. El usuario elegirá aquellas restricciones que mejor encajen con su proyecto.
4. Ver priorizadas las técnicas y métodos que la herramienta recomienda.
  - a. El usuario, en función de las selección de restricciones anterior, verá la lista priorizada de técnicas y métodos.
5. Adaptar la aplicación en base a su contexto organizacional para afinar la lista de técnicas y métodos recomendados.
  - a. El usuario podrá modificar la lógica de la aplicación para introducir nuevas técnicas, métodos y restricciones o bien modificar la fórmula para realizar la priorización.

### 3.1.5 Nombre de la aplicación.

Se planteó en dar un nombre a la aplicación que fuera fácil de recordar para los usuarios. Tras realizar un *brainstorming* se pensó que el nombre de la aplicación debía cumplir los siguientes objetivos: Fácil de pronunciar y recordar; debía estar formado por entre una y dos palabras; y descriptivo (no crear marcas ni nombres no conocidos).

Entre los nombres que se pensaron se eligieron tres candidatos:

- UCD Tool.
- Usability Tool.
- UCD Picker.

Finalmente se apostó por **UCD Picker**, en alusión a las máquinas recreativas con forma de cabina que guardan en su interior regalos, como por ejemplo peluches, u otros objetos esperando a ser recogidos por una grúa mecánica y la habilidad del usuario que maneja la máquina para lograrlo (ver Fig. 19).



Fig. 19. Máquina para capturar peluches (en inglés *teddy picker*).

Se eligió este nombre porque se pensó que iba a sorprender al usuario la analogía descrita y este factor iba ser relevante para cumplir los objetivos planteados anteriormente. Incluso se pensó en crear un logotipo que fuese una representación abstracta de una de estas máquinas, para reforzar esta metáfora.

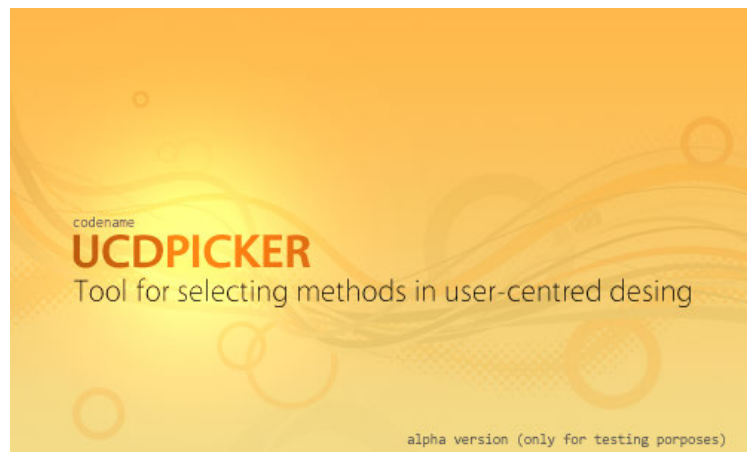


Fig. 20. *Splash screen* (pantalla de carga) de UCD Picker.

### 3.1.6 Wireframes.

Los *wireframes* (prototipo en papel, en español) son una representación esquemática de la interfaz de usuario del sistema, son un medio para mostrar la arquitectura, contenido y comportamiento del sistema sin necesidad de realizar una implementación del mismo. Es una de las técnicas de usabilidad mas empleadas por su bajo coste y efectividad.

En el desarrollo del proyecto ha sido una de las técnicas más empleadas. Sirve tanto para conceptualizar y diseñar la interacción del usuario con el sistema como para evaluar la usabilidad.

En la Fig. 21 se puede ver el primer *wireframe* que se realizó de la aplicación.

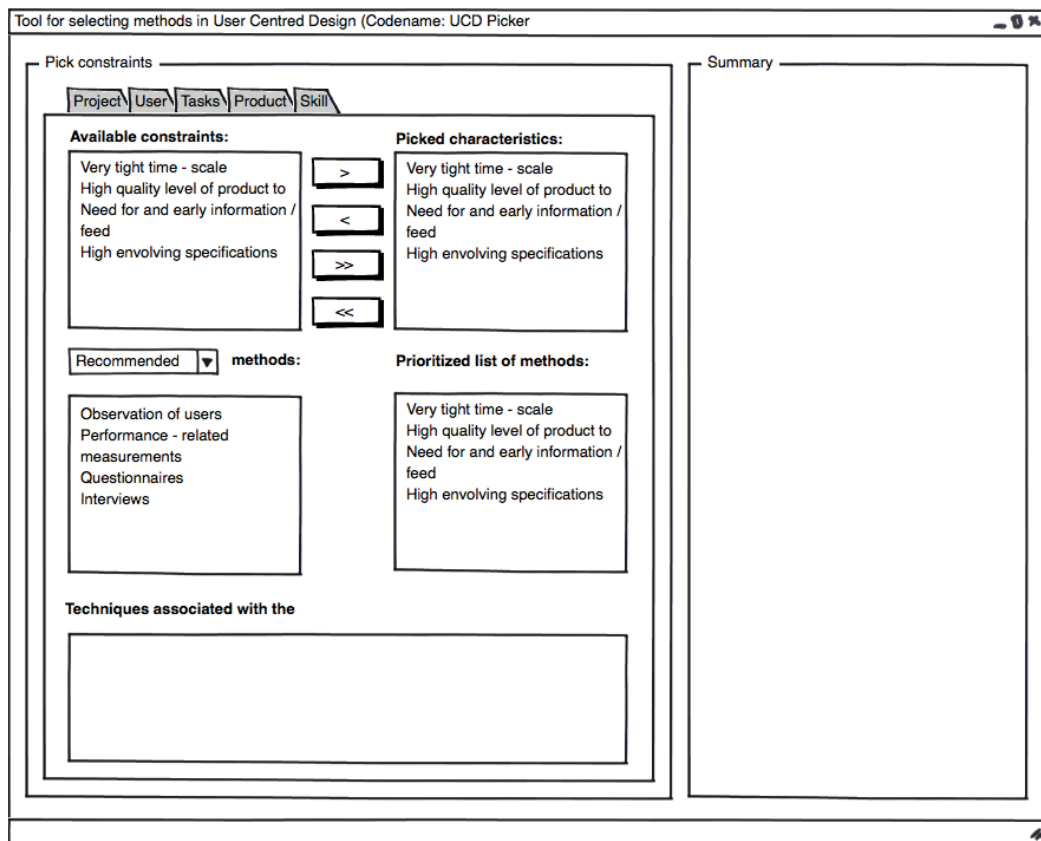


Fig. 21. *Wireframe* de la pantalla principal de la primera versión de la aplicación.

La pantalla que se muestra en la Fig. 21 corresponde a la vista en la que el usuario ya ha seleccionado las etapas del ciclo de vida en las que ha planificado llevar a cabo métodos de usabilidad. Como se puede observar, la idea inicial era que la selección de restricciones pudiera ser realizada en una única vista, de forma que, el usuario, en función de las restricciones seleccionadas, viera simultáneamente cómo varía la lista priorizada de métodos.

A la derecha, aunque no llegó a implementarse, se previó un espacio para mostrar un sumario que recogiera la lista priorizada de técnicas y métodos asociadas a cada etapa del ciclo de vida, así como las buenas prácticas que se recomiendan para llevarlas a cabo.

Tras la realización de varios *wireframes* se implementó un prototipo funcional de la aplicación.



### 3.1.7 Desarrollo del prototipo.

Se optó por crear un prototipo que implementase las tareas más relevantes (aquellas relacionadas con la selección de restricciones y priorización de técnicas y métodos), para tener, en un corto periodo de tiempo, una muestra de cómo sería la experiencia de usuario (ver Fig. 22).

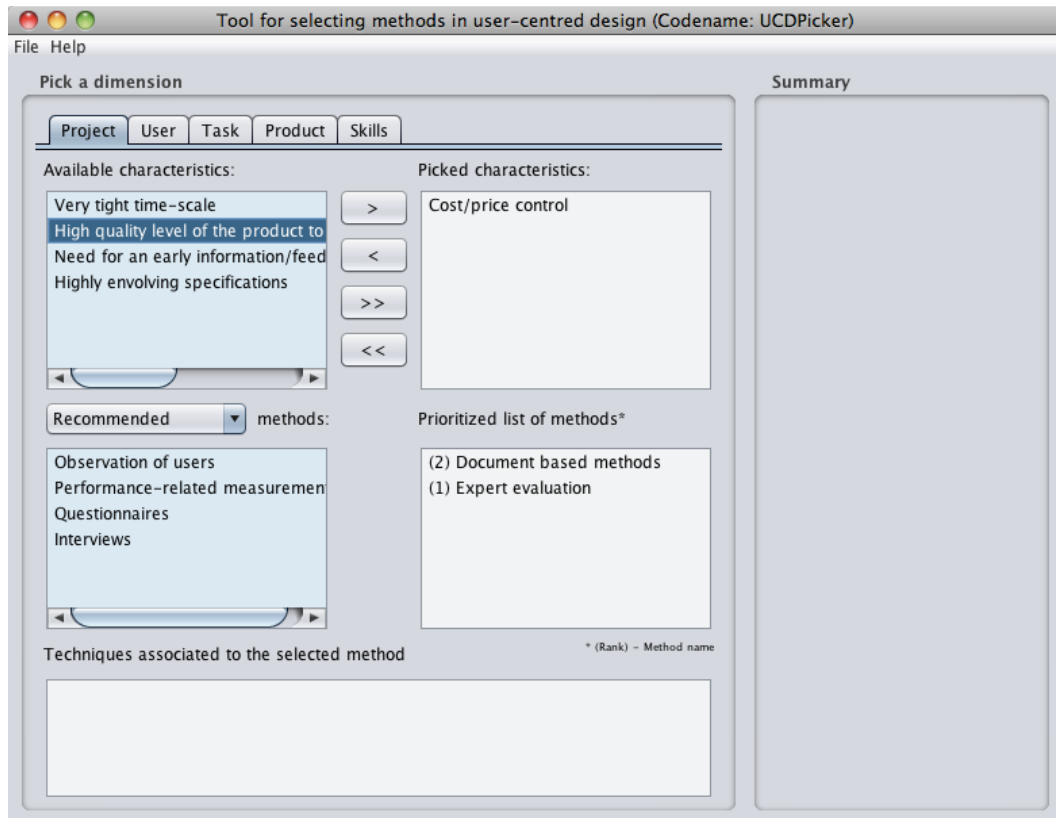


Fig. 22. Captura de pantalla de UCD Picker (selección de restricciones).

El hecho de ver funcionando el prototipo arrojó nuevos requisitos y oportunidades. Por ejemplo, se pensó que la aplicación debería ser más guiada, es decir, en lugar de manejar toda la información en una única vista, un proceso paso a paso sería mucho más intuitivo y fácil de manejar por los usuarios. Y sobre todo y como resultado de una evaluación realizada por Bevan y Ferré, se determinó que la aplicación tendría mucho más potencial si el marco en el que se utilizase fuera el navegador web.

## 3.2 Segunda iteración.

La segunda iteración destaca principalmente por el cambio de tecnología de la aplicación. Se llegó a la conclusión de que una aplicación de escritorio iba a limitar la evolución del sistema a nivel de escalabilidad, distribución y mantenimiento. Se pensó que un marco como la web sería más enriquecedor para el proyecto.

Entre las ventajas que una aplicación web tiene frente a una aplicación de escritorio, fundamentalmente una de ellas es la distribución; poder ejecutar el sistema desde cualquier navegador instalado en un ordenador o dispositivo móvil, es una característica muy atractiva, al igual que la disponibilidad y el hecho de tener siempre actualizado el sistema de forma transparente al usuario.

El reto de encontrar la tecnología mas conveniente para afrontar este cambio fue interesante ya que se debía cumplir un requisito nuevo, reutilizar el código fuente escrito en Java del primer prototipo. Por lo tanto había que buscar una tecnología de desarrollo similar (basada en Java) que permitiera crear un sistema web con las ventajas de un sistema o aplicación de escritorio y que permitiera su ejecución de forma aislada o local (*stand-alone*) sin realizar ninguna conexión con otros sistemas remotos tales como bases de datos o servidores de aplicaciones.

Mas adelante, en el capítulo 4, se detalla en profundidad como fue llevada a cabo esta transición y la solución elegida para cumplir las premisas anteriores.

Esta segunda iteración destaca además por el replanteamiento de la arquitectura de información y de los contenidos, la revisión de los requisitos inicialmente planteados, así como la creación de un escenario que ayudase a comprender, mediante un caso real, la interacción del usuario con el sistema.

Otra de las claves de esta iteración fue la realización de un test de usabilidad con usuarios, el cual arrojó importantes aspectos a considerar para las siguientes iteraciones.

### 3.2.1 Revisión de la conceptualización.

Ante el nuevo paradigma tecnológico y con el objetivo de crear una aplicación más potente se realizó, nuevamente, una tormenta de ideas (ver Fig. 23).



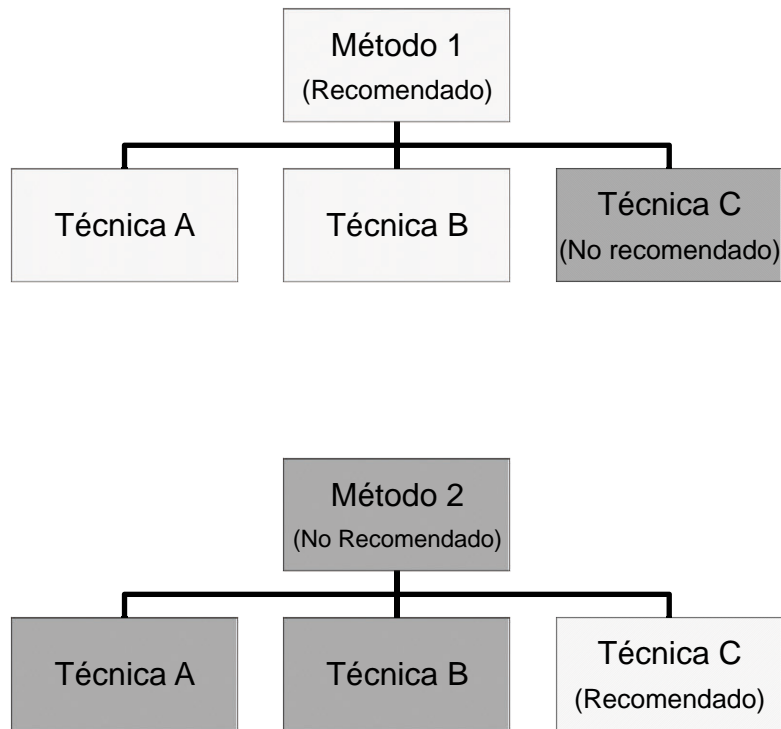


Fig. 24. La técnica C puede ser recomendada o no independientemente del método lo sea o no.



En la Fig. 25 se muestra un esquema de la arquitectura de información a la que se llegó al finalizar el *brainstorming*. El punto de partida de la arquitectura es la etapa del ciclo de vida en la que el usuario quiere priorizar técnicas.

Asociada a cada etapa se enumeran las buenas prácticas que en términos de usabilidad se recomiendan llevar a cabo para minimizar los riesgos y maximizar los beneficios del proyecto.

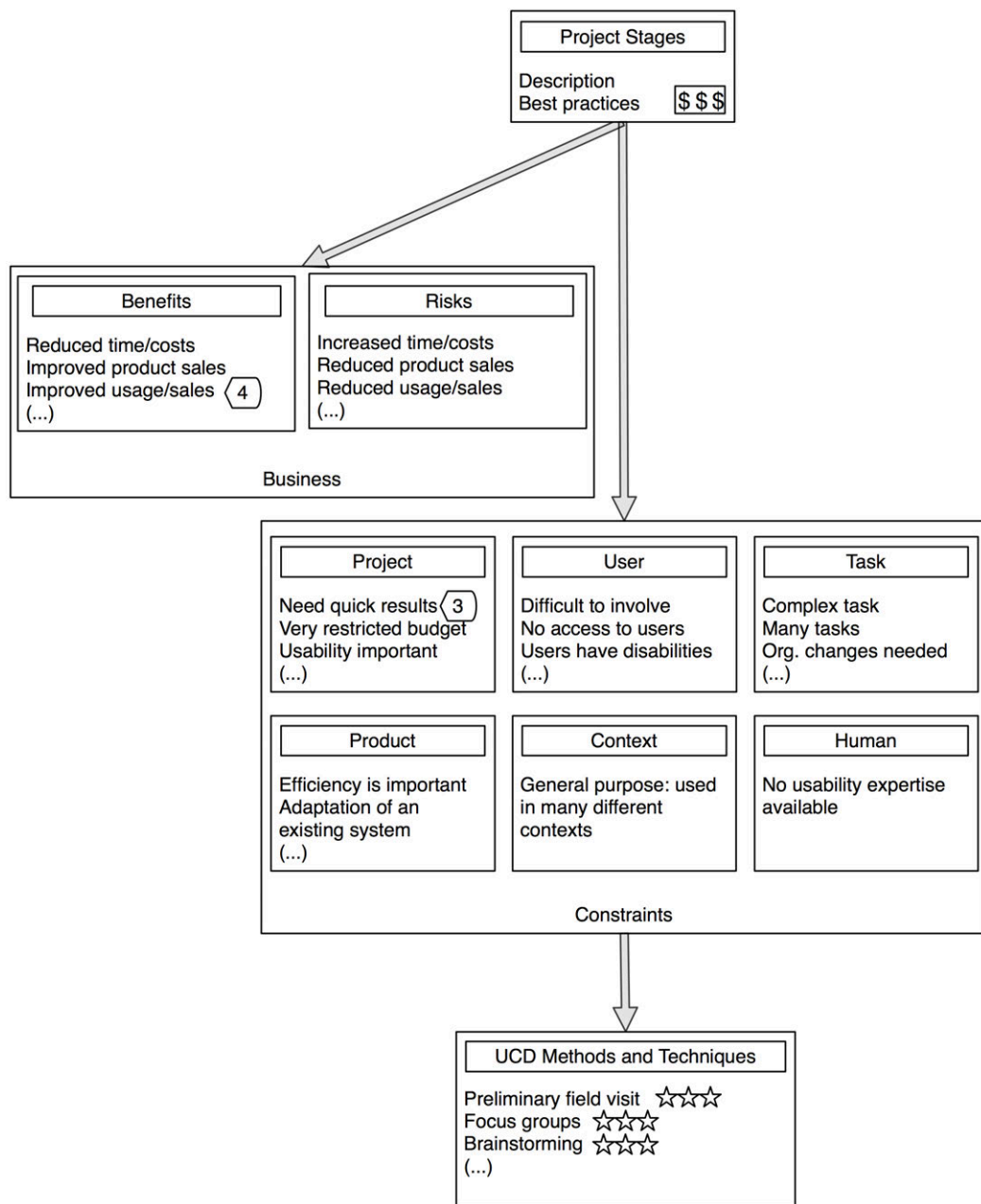


Fig. 25. Diagrama conceptual de los contenidos de la aplicación.

Cada etapa del ciclo de vida del proyecto es susceptible a ser afectada por restricciones relacionadas con los usuarios, las tareas, el producto a desarrollar, el contexto y la experiencia de las partes involucradas en el desarrollo.

El árbol de la Fig. 25 termina con la priorización de las técnicas recomendadas en función de las restricciones y las etapas que el usuario determina.

### 3.2.2 Revisión de las personas y análisis de tareas.

En la primera iteración se identificaron tres personas. En esta iteración se decidió agrupar conceptualmente las dos primeras (Sara y Carmen), por sus similitudes, pero sin perderlas como personas independientes. La razón fue que la mayoría de tareas las comparten siendo sus objetivos similares.



Fig. 26. Agrupación de personas.

Por tanto, se determinaron dos grandes metas, o dos grandes tareas a desempeñar en el sistema:

- Planificar técnicas de usabilidad. Objetivo principal de Carmen y Sara.
- Planificar el proyecto en términos DCU. Objetivo principal de Steven.

En el siguiente diagrama se puede ver esta diferencia. Mientras que unos usuarios buscan una selección de métodos recomendados para aplicar en una determinada etapa del ciclo de vida, otros además buscan que esa priorización sea en términos de negocio, es decir, priorizar las técnicas que maximizan los beneficios y minimizan los riesgos.

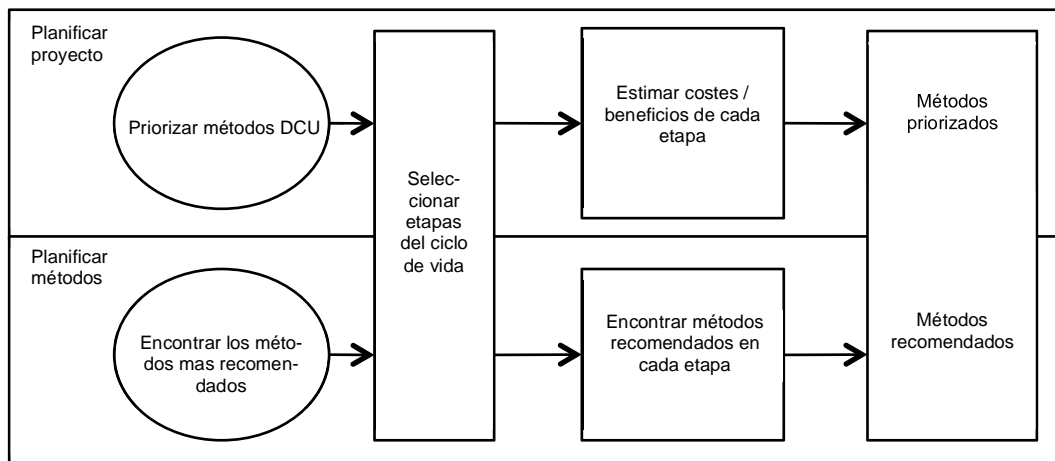


Fig. 27. Diagrama de interacción y flujo de tareas.

### 3.2.3 Creación de un escenario.

Con el objetivo de analizar la interacción de los usuarios con el sistema, para el desempeño de sus objetivos y necesidades, se empleó la técnica de los **escenarios de uso**. Los escenarios son descripciones de uno o más usuarios interactuando con el sistema para alcanzar una serie de metas en un determinado contexto de uso (Cooper, 2004).

A continuación se describe un escenario protagonizado por Carmen:

*Carmen es consultora junior en experiencia de usuario en Usertic, compañía española dedicada al desarrollo de software con enfoque DCU. Actualmente, Carmen está trabajando en un proyecto llamado “Visiguide”, que consiste en el desarrollo de un sistema para guiar a los visitantes de una institución a través de distintas salas de exposiciones.*

*Carmen está encargada de seleccionar las técnicas de usabilidad y actividades del proceso DCU más convenientes para el proyecto, teniendo como premisa principal que el presupuesto es muy bajo. Para realizar este estudio Carmen utiliza UCD Picker.*

*Debido a que el presupuesto es bastante limitado, Carmen tiene especial interés en priorizar las técnicas en base a esta premisa. Como el proyecto acaba de empezar, Carmen decide primero investigar las actividades de la etapa de conceptualización de la aplicación, por ello marca en la aplicación la etapa de conceptualización “1. Concept”.*

*A continuación, Carmen revisa la lista de técnicas y métodos recomendados para la etapa elegida así como para las subetapas en las que se divide (previsión de oportunidades y alcance del sistema).*

*Con el fin de ajustar la lista de métodos según las características de “Visigui-de” comienza a marcar las restricciones que mejor lo describen:*

*Contantes relacionadas con el proyecto en general:*

- ☒ *Se requieren resultados rápidos.*
- ☒ *No existe una especificación detallada.*
- ☒ *Presupuesto muy ajustado.*

*Restricciones relacionadas con los usuarios:*

- ☒ *La mayoría de usuarios no han interactuado con sistemas similares.*

*En base al criterio introducido por Carmen la aplicación le devuelve la siguiente lista priorizada:*

*Técnicas recomendadas:*

- *Tormenta de ideas mediante dibujos.*
- *Tormenta de ideas.*

*Técnicas apropiadas:*

- *Estudios de campo preliminares.*
- *Consulta con las partes interesadas.*
- *Estudios con fotografías.*
- *Análisis del contexto de uso.*



*En base a los resultados, Carmen comienza por la planificación de una sesión de tormenta de ideas. Mas tarde, según avanza la conceptualización del proyecto, Carmen decide recurrir nuevamente a la aplicación para planificar las técnicas a utilizar en las siguientes etapas del proyecto.*



La elaboración de este escenario ayudó a comprender en que contexto tendría lugar el uso de la aplicación: Probablemente compañías de desarrollo de software con equipos especializados en usabilidad mayoritariamente formados por consultores júnior, que necesitan planificar las actividades del proceso DCU para el desarrollo de un proyecto en el menor tiempo posible. Debido a que los integrantes de los equipos no tienen gran experiencia en el ámbito de la usabilidad, necesitan una aplicación que les ayude a planificar las técnicas a emplear.

### 3.2.3 Revisión del nombre de la aplicación.

Se llegó a la conclusión que el nombre elegido en la iteración anterior, UCD Picker, no reflejaba completamente el sentido y propósito de la aplicación, de hecho, podría incluso hacer creer al usuario que la selección de técnicas y métodos se hacía de forma aleatoria (por el comportamiento de la máquina a la que hace alusión).

Se necesitaba un nombre que reflejase los dos principios básicos que diferencian a esta aplicación de otras: La **planificación** y **priorización** de métodos y técnicas usabilidad.

Se realizó una nueva sesión de *naming* (búsqueda del nombre a una marca, en español) para definir uno más adecuado (ver Fig. 29). Los nombres candidatos fueron:

- Usability Planner.
- Usability Methods Planner.
- Uplantool.
- U-plan.
- U-Planner.

Finalmente se eligió **Usability Planner**.



Fig. 28. Primer diseño del logotipo de la aplicación.

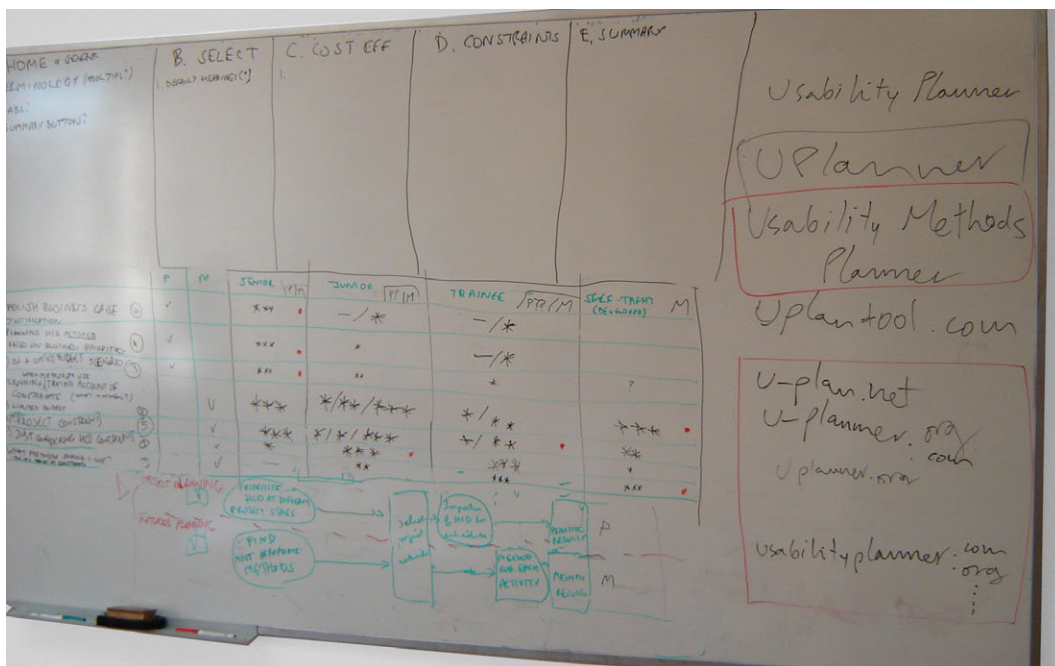


Fig. 29. Imagen de la pizarra tomada de la sesión de *naming*.

### 3.2.4 Producción de una nueva solución de diseño.

Al igual que en la iteración anterior se realizaron prototipos en papel antes de pasar a la etapa de desarrollo. En esta iteración se realizaron hasta 6 prototipos de baja fidelidad.

Se realizó un test de usabilidad con usuarios. En el siguiente apartado se detallan los resultados del test.

A continuación, en la Fig. 30 y siguientes, se muestra el último *wireframe* que se realizó en esta iteración.

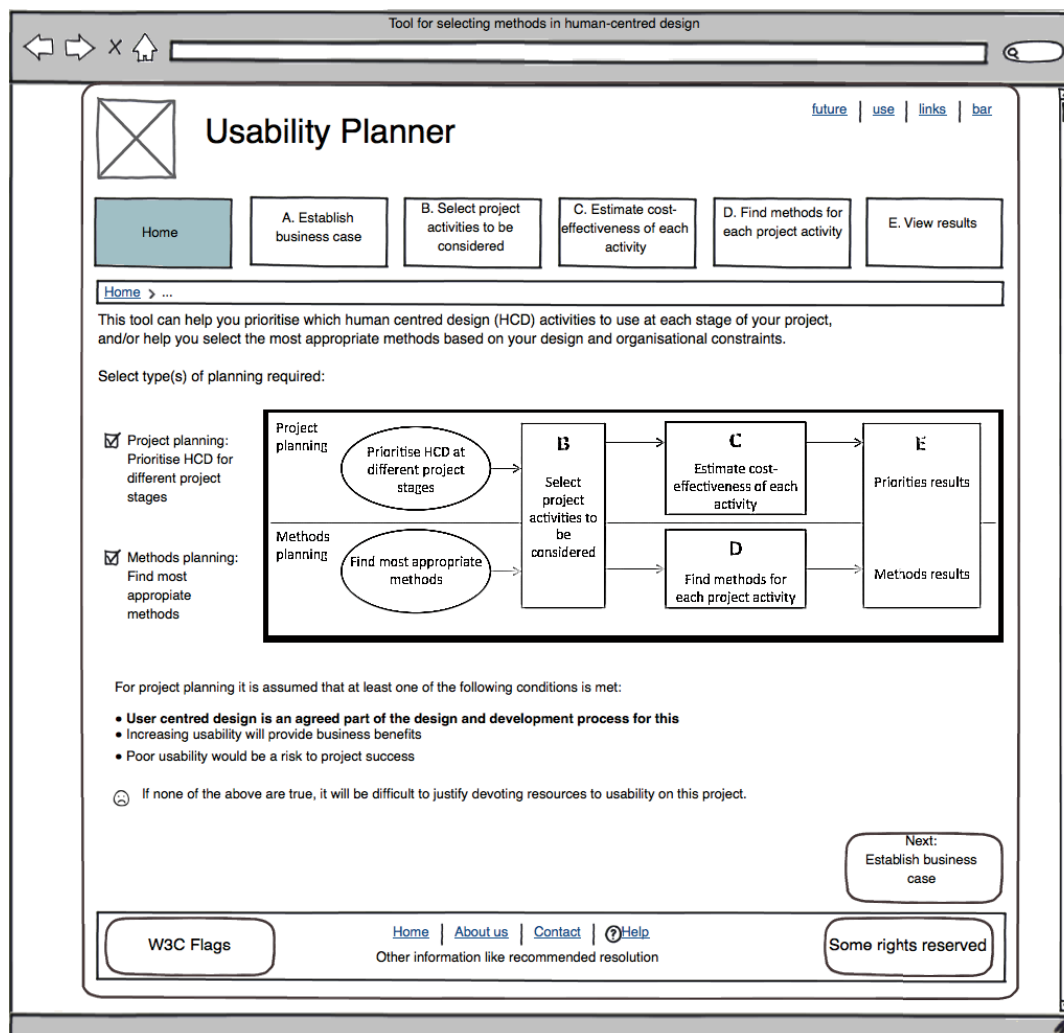


Fig. 30. Pantalla de inicio de la aplicación (versión 6).

Como se puede observar en la Fig. 30, en el encabezado de la página se encuentra el logotipo y nombre de la aplicación, unos enlaces para en un futuro cambiar el idioma, la barra de navegación y el camino de migas de pan (también llamado *breadcrumb*, en inglés) para indicar al usuario en qué punto de la jerarquía de páginas se encuentra.

Se optó por una barra de navegación horizontal debido a que no se previó que los apartados de la aplicación creciesen en un futuro y para transmitir al usuario el carácter “paso a paso” o guiado de la herramienta.

En la parte central de la ventana se explica el propósito de la herramienta y se da a elegir al usuario entre dos tipos de planificación (que responden a los objetivos de los perfiles de usuario identificados anteriormente).

Tras una breve recomendación y para enfatizar el carácter guiado de la herramienta se sitúa un botón en la parte inferior derecha para ir a la siguiente pantalla (ver Fig. 31).

The screenshot shows a web browser window titled "Tool for selecting methods in human-centred design". The main content area is titled "Usability Planner" and features a navigation bar with links: [future](#), [use](#), [links](#), and [bar](#). Below the navigation bar is a horizontal menu with five buttons: "Home", "A. Establish business case" (highlighted in blue), "B. Select project activities to be considered", "C. Estimate cost-effectiveness of each activity", "D. Find methods for each project activity", and "E. View results". A breadcrumb trail shows "Home > Establish business case". The main section is titled "A. Establish business case to justify using resources for usability" and includes a sub-header "Filling in this page is optional, but will help you if you have any doubts about the business case for usability." Below this is a form with a legend: "Select + (agree), 0 (neutral) or - (disagree) for each statement". The form contains three radio button options: "User centred design is an agreed part of the design and development process for this", "Increasing usability will provide business benefits", and "Poor usability would be a risk to project success". A checkbox option is also present: "If none of the above are true, it will be difficult to justify devoting resources to usability on this project." A paragraph of text follows: "Review the potential benefits and risks to identify which could apply to your project. This will help you prioritise which methods to use in step C." A "Next" button is located at the bottom right, labeled "Next: Select project activities to be considered". The footer contains a "W3C Flags" button, a navigation bar with links: [Home](#), [About us](#), [Contact](#), and [Help](#), and a "Some rights reserved" button.

Fig. 31. Pantalla para la selección del tipo de modelo de negocio.

En la pantalla de la Fig. 31, el usuario puede justificar el uso de recursos para usabilidad en su proyecto. Realmente se trata de una pantalla para que el usuario reflexione si el uso de técnicas y métodos de usabilidad en su proyecto es justificable.

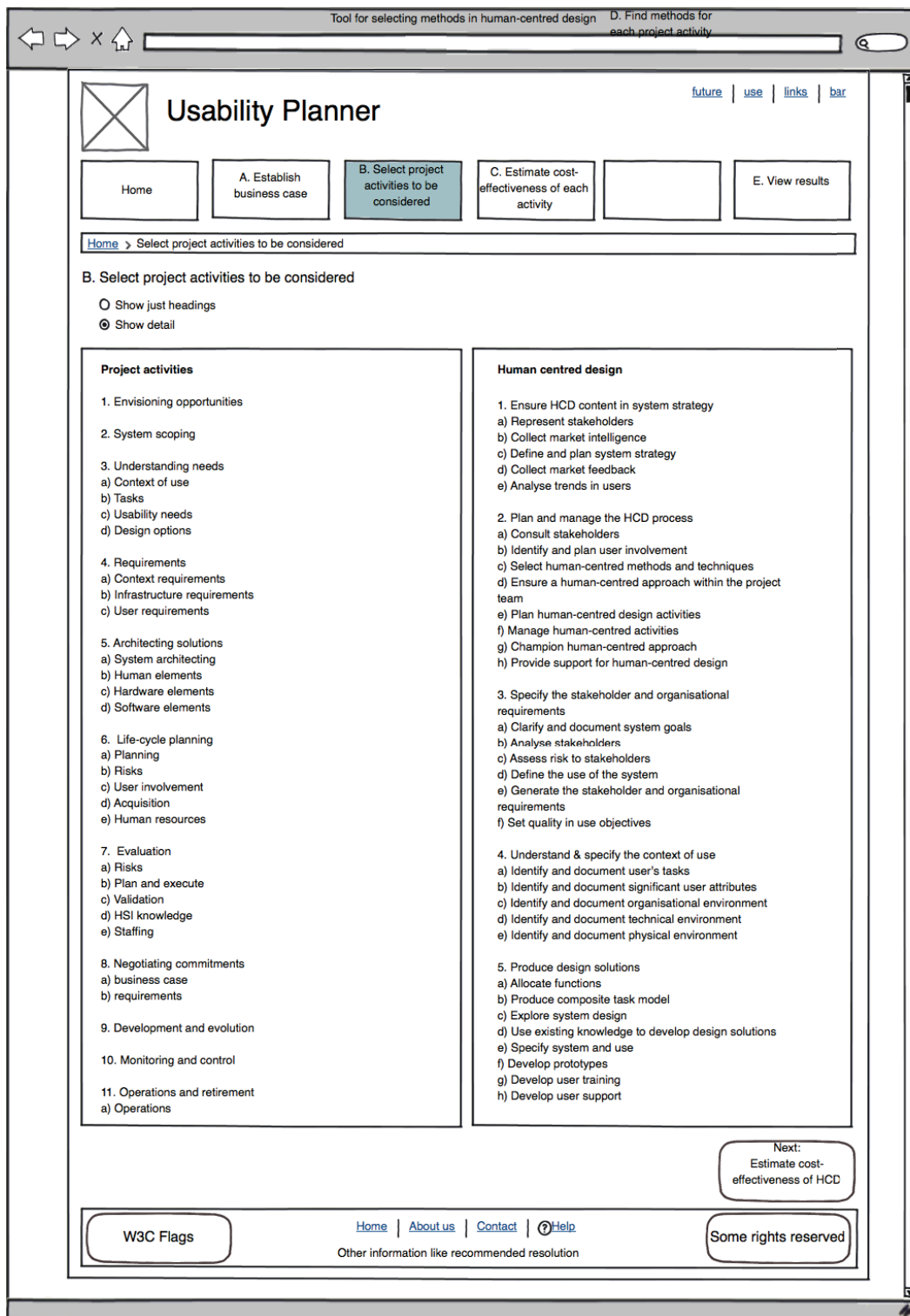


Fig. 32. Pantalla para la selección de las etapas del ciclo de vida.

En la pantalla de la Fig. 32 está representada la selección de las etapas del ciclo de vida. Se ofrecen dos opciones, mostrar el listado únicamente con la etapa principal o bien mostrar además las sub-etapas de cada una de ellas.

Tool for selecting methods in human-centred design

future | use | links | bar

# Usability Planner

Home | A. Establish business case | B. Select project activities to be considered | C. Estimate cost-effectiveness of each activity | D. Find methods for each project activity | E. View results

Home > Estimate cost-effectiveness of each activity

## C. Estimate cost-effectiveness of each activity

Envisioning opportunities

High.....Low How likely is it that using HCD methods (such as those shown below) for Envisioning opportunities would be cost-effective?

System scoping

Requirements

...

### HCD activities (best practices)

Identify expected context of use of systems (forthcoming needs, trends and expectations)

Analyze the system concept (to clarify objectives, their viability and risks)

### Potential HCD methods and techniques

You can deselect any which are inappropriate. (The methods will be prioritised in the next step.)

- Future workshop
- Preliminary field visit
- Photo surveys
- Focus groups
- Simulations of future use environments

The cost effectiveness of using HCD methods for Envisioning opportunities will depend on the relevance of potential business benefits and risk shown below

### Potential business benefits

- Reduced development costs
- Web site usability: improved web sales
- Product usability: improved product sales
- Improved productivity: benefits to purchasing organisation
- Reduced support and maintenance costs

### Potential risks

- Increased development costs to produce an acceptable system
- Web site usability: poor web
- Product usability: poor product sales
- Poor productivity: risks to purchasing organisation
- Increased support and

Next: Find methods for each project activities

W3C Flags

Home | About us | Contact | Help

Other information like recommended resolution

Some rights reserved

Fig. 33. Pantalla de estimación de beneficios y riesgos por cada actividad.

En la Fig. 33 se puede ver la representación del siguiente paso. El objetivo de esta pantalla es dar a conocer al usuario las buenas prácticas que se recomiendan realizar en cada una de las etapas seleccionadas en el paso anterior.

El contenido principal de la pantalla queda dividido en cuatro bloques:

- Bloque superior izquierdo: Contiene el listado de las buenas prácticas asociadas a las etapas elegidas. Por defecto se muestran los titulares de cada una. No obstante, si el usuario pincha en cada titular se despliega información adicional detallada de cómo llevar a cabo cada práctica.
- Bloque superior derecho: Contiene el listado de técnicas de usabilidad recomendadas (en general) en las etapas seleccionadas. Si el usuario pincha en el título de cada técnica se despliega una breve descripción de la técnica.
- Bloque inferior izquierdo: Muestra un listado con los potenciales beneficios de la aplicación de técnicas de usabilidad. Si el usuario pulsa sobre cada titular, el sistema despliega información adicional.
- Bloque inferior derecho: Muestra un listado con los potenciales riesgos de la no aplicación de técnicas de usabilidad. Si el usuario pulsa sobre cada titular, el sistema muestra información adicional.

Al igual que en las anteriores pantallas se ha situado un botón para avanzar al siguiente paso en la esquina inferior derecha.

La siguiente pantalla (ver Fig. 34) representa la selección de las restricciones que se dan en un determinado proyecto y en función de las cuales el sistema ofrecerá una priorización de técnicas.

El contenido principal de la pantalla se divide en dos bloques:

- Bloque izquierdo: Listado de tipos de restricciones. El usuario puede especificar el grado de conformidad con cada una de ellas, de ahí que se hayan situado tres selectores (*radio buttons*, en inglés) para expresar una conformidad baja, neutra o alta con el significado de la restricción que acompaña.



- Bloque derecho: Muestra el listado de las técnicas de usabilidad recomendadas en función de la selección de restricciones realizada por el usuario. Acompañando a cada técnica, una sucesión de hasta cinco estrellas indica el nivel de recomendación.

The screenshot shows a web browser window titled "Tool for selecting methods in human-centred design". The main content area is titled "Usability Planner" and features a navigation bar with buttons: Home, A. Establish business case, B. Select project activities to be considered, C. Estimate cost-effectiveness of each activity, D. Find methods for each project activity (highlighted), and E. View results. Below the navigation bar is a breadcrumb trail: Home > Find methods for each project activity. The main section is titled "D. Find methods for each project activity". On the left, there is a sidebar with a tree view containing "Envisioning opportunities", "System scoping", "Requirements", and "...". The main content area has two columns. The left column contains a list of constraints grouped into six categories: 1. Project characteristics, 2. User issues, 3. Task characteristics, 4. Product characteristics, 5. Human factors expertise, and 6. Contexts of use. Each category has a set of radio buttons and a dropdown arrow. The right column is titled "Recommended" and contains a list of recommended methods with star ratings and names: Future (5 stars), Preliminary (5 stars), Focus (4 stars), Photo surveys (4 stars), and Simulations of future use (2 stars). Below the recommended methods is a button labeled "Next: View results". At the bottom of the page, there is a footer with "W3C Flags", a navigation bar with "Home", "About us", "Contact", and "Help", and a "Some rights reserved" button.

Tool for selecting methods in human-centred design

Usability Planner

Home | A. Establish business case | B. Select project activities to be considered | C. Estimate cost-effectiveness of each activity | D. Find methods for each project activity | E. View results

Home > Find methods for each project activity

D. Find methods for each project activity

Envisioning opportunities  
System scoping  
Requirements  
...

☒ Adjust the project constraints to see the recommended methods ☐ Select a method to see the constraints for that method

Click for more information.  
Select + (agree), 0 (neutral) or - (disagree)  
+ 0 -

1. Project characteristics

☐ ☐ ☐ Need quick results ▾  
☐ ☐ ☐ Very restricted budget ▾  
☐ ☐ ☐ Usability important ▾  
☐ ☐ ☐ Rapid feedback required ▾  
☐ ☐ ☐ Uncertain specification ▾

2. User issues

☐ ☐ ☐ Difficult to involve users ▾  
☐ ☐ ☐ No access to users ▾  
☐ ☐ ☐ Some users have disabilities ▾

3. Task characteristics

☐ ☐ ☐ Complex task ▾  
☐ ☐ ☐ Many tasks ▾  
☐ ☐ ☐ Mostly first time users ▾  
☐ ☐ ☐ Safety or business critical system ▾  
☐ ☐ ☐ Organisational changes needed ▾

4. Product characteristics

☐ ☐ ☐ Time or accuracy is important ▾  
☐ ☐ ☐ Adaptation of an existing system ▾  
☐ ☐ ☐ A well understood product ▾  
☐ ☐ ☐ Customisable product ▾

5. Human factors expertise

☐ ☐ ☐ No human factors expertise available ▾

6. Contexts of use

☐ ☐ ☐ Used in many different contexts ▾

Recommended

The more stars the better.  
Select a method to see the constraints for that method

★★★★★ ☐ Future  
★★★★★ ☐ Preliminary  
★★★★ ☐ Focus  
★★★★ ☐ Photo surveys  
★★ ☐ Simulations of future use

Click a method name to see more information

☐ Specify different constraints for each project activity

Next: View results

W3C Flags

Home | About us | Contact | Help

Other information like recommended resolution

Some rights reserved

Fig. 34. Pantalla para la selección de restricciones.

Por último, en la Fig. 35 se puede ver el *wireframe* del último paso de la secuencia, en la que el usuario puede ver un resumen con los resultados que ha obtenido en los pasos anteriores.

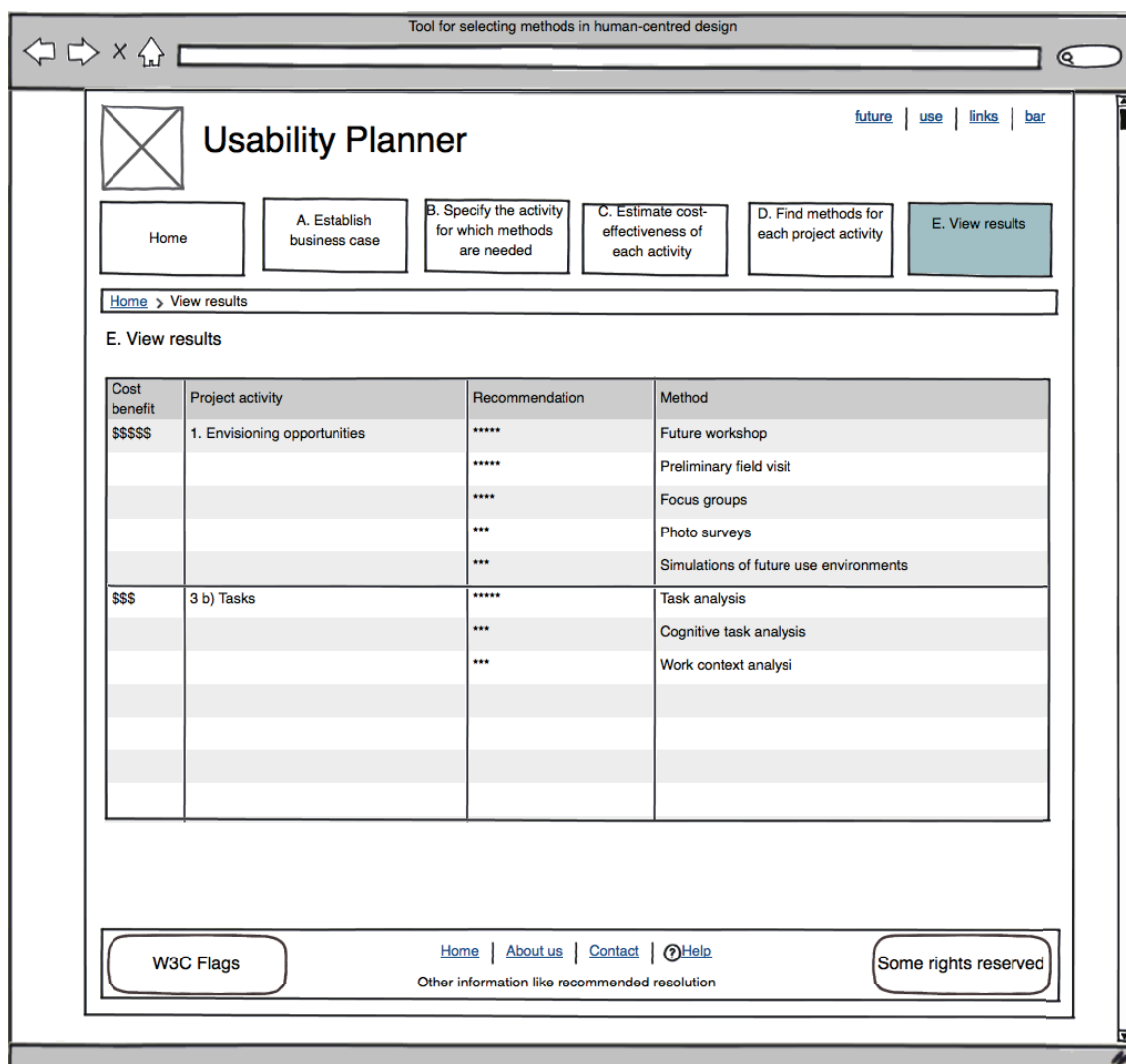


Fig. 35. Pantalla del paso final.

Mediante una tabla se muestra el coste/beneficio, las etapas del ciclo de vida y los métodos y técnicas de usabilidad junto con la priorización realizada. La principal utilidad de esta pantalla es que el usuario pueda tener en una única vista su plan de usabilidad.

### 3.2.5 Realización de un test de usabilidad.

El test de usabilidad con usuarios (también denominado test de usuario) es una técnica que permite comprobar cómo se desenvuelve un grupo de usuarios al realizar una serie de tareas con el sistema. La sesión se prepara de forma que cada usuario, individualmente, en compañía de un evaluador, complete unas tareas predefinidas.

Durante el transcurso de la prueba se utiliza el protocolo de **pensamiento en voz alta** (*thinking aloud*, en inglés) que consiste en pedir al usuario que verbalice sus pensamientos durante el transcurso de la prueba de forma que el evaluador pueda recogerlos.

Para la prueba se seleccionaron 8 estudiantes de Máster cursando la asignatura de Diseño de la Interacción, de edades comprendidas entre los 23 y 32 años y distintas nacionalidades.

Se les plantearon únicamente dos tareas:

1. “¿Qué técnicas de usabilidad emplearías en tu proyecto en la sub-etapa de análisis del contexto de uso? ¿Qué buenas prácticas se recomiendan para su aplicación?”
2. “¿Qué técnicas de usabilidad utilizarías en el caso de que tu presupuesto sea ajustado y tengas un difícil acceso a los usuarios para la realización del proyecto?”

Normalmente las pruebas suelen ir acompañadas de unas métricas, como por ejemplo, el número de clics que el usuario realiza hasta resolver con éxito la tarea, el tiempo empleado en completarla con éxito, el camino seguido, etc. En este caso, no se tuvieron en cuenta debido a la sencillez de la prueba. Todo el foco de atención se puso en observar a los usuarios y anotar sus reacciones y comentarios (ver Fig. 38).

Para la realización del test se utilizó el prototipo en papel realizado en esta iteración.

A continuación se enumeran los comentarios más frecuentes que realizaron los usuarios así como las dificultades que tuvieron a la hora de realizar cada tarea:

- Todos los usuarios, en algún momento, expresaron que la interfaz tenía demasiado texto. Veían demasiada información y no leían las indicaciones.
- En la primera pantalla (ver Fig. 30), 7 de los 8 participantes tuvieron dificultades para comprender la diferencia entre los dos tipos de planificación: Planificación a nivel de proyecto y planificación nivel de métodos (ver Fig. 36).

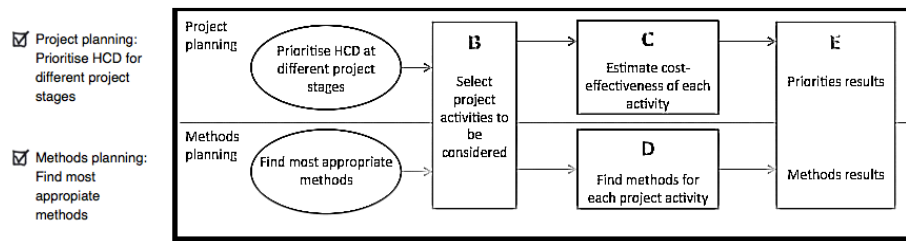


Fig. 36. Selección del tipo de planificación.

- 7 de los 8 participantes fueron al paso “A. *Establish business case*” (ver Fig. 31) cuando realmente este apartado no es necesario para completar con éxito la tarea. Los 7 participantes tuvieron dificultades para continuar desde ese punto, a pesar de que era opcional y así quedaba reflejado en la explicación de la pantalla.
- 6 de los 8 participantes expresaron que no entendían la diferencia entre los dos listados de actividades en el paso B (ver Fig. 32) y dudaron en cual de los dos debían realizar la selección.
- Todos los usuarios supieron identificar la lista de técnicas y métodos, así como las buenas prácticas asociadas que el sistema devuelve en el paso C (ver Fig. 33). No obstante, expresaron que no entendían el resto de información de la pantalla (en relación a la información relacionada con potenciales beneficios y riesgos de negocio).
- En la tarea 2, todos los participantes coincidieron en que la triple selección (ver Fig. 34 y Fig. 37), para expresar el grado de conformidad con cada restricción era complicado. Sugirieron tener un único selector con dos estados (*checkbox*).

1. Project characteristics
- ☒ ☒ ☐ Need quick results ▽
  - ☒ ☒ ☐ Very restricted budget ▽
  - ☒ ☒ ☐ Usability important ▽
  - ☒ ☒ ☐ Rapid feedback required ▽
  - ☒ ☒ ☐ Uncertain specification ▽

Fig. 37. Triple selección para cada restricción.

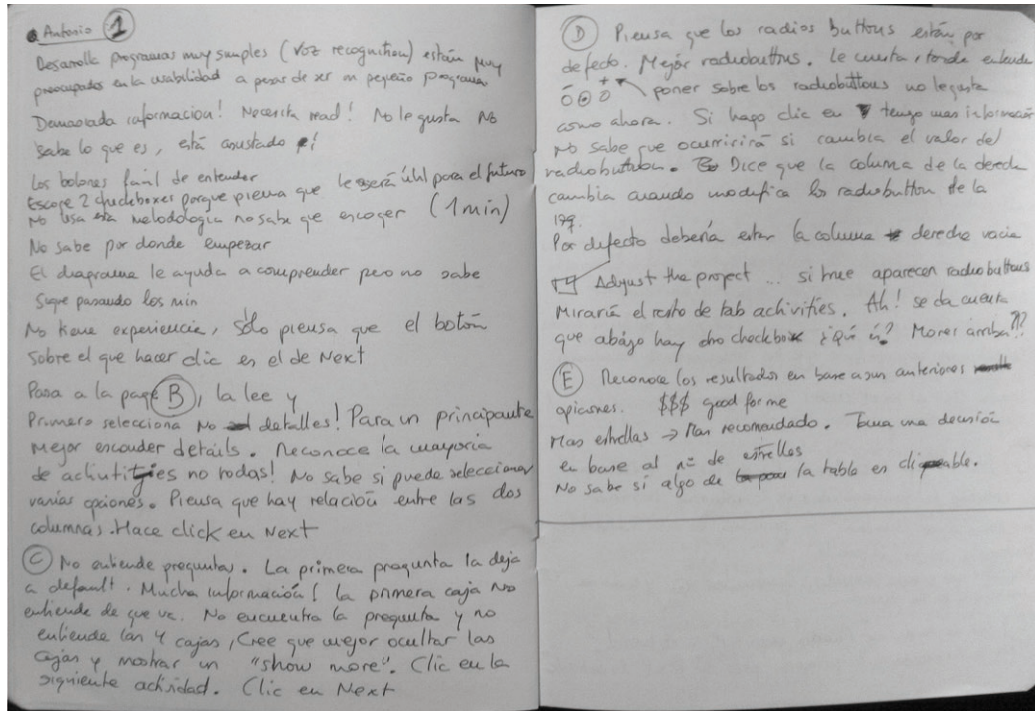


Fig. 38. Hoja con las anotaciones durante el test con uno de los participantes.

### 3.2.5 Desarrollo de la solución de diseño.

Con toda la información recopilada en esta etapa se creó un nuevo *wireframe* como paso previo para el desarrollo del prototipo.

A continuación, ver Fig. 39 y siguientes, se muestran las pantallas de la implementación de esta versión con los cambios realizados en base a los resultados de las pruebas anteriores.

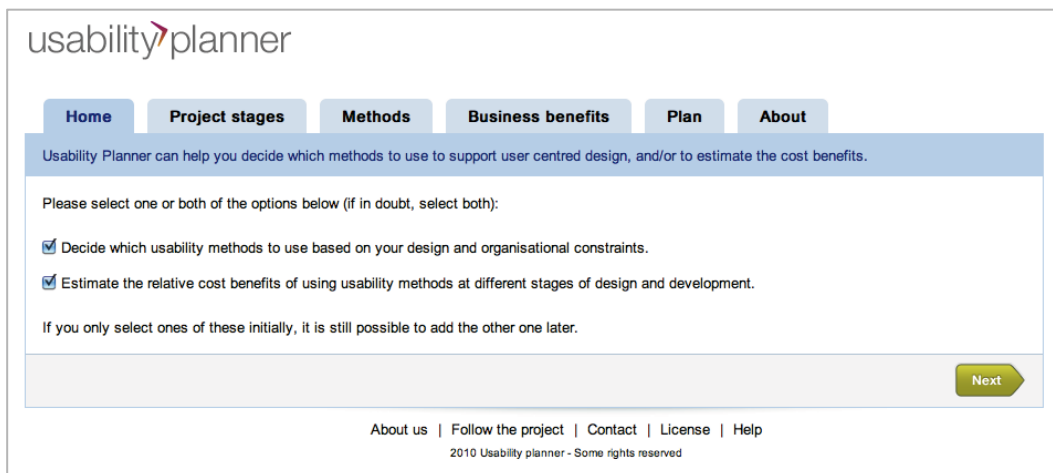


Fig. 39. Pantalla de inicio de la aplicación.

El primer cambio que destaca en la implementación con respecto al *wireframe* descrito anteriormente es la modificación del flujo de navegación. En la implementación (ver Fig. 39) la parte de beneficios de negocio se condensa en un solo paso, el tercero, en lugar de tener dos pasos (el A y C propuesto en el *wireframe* - Fig. 30).

El comportamiento del selector del tipo de planificación (ver Fig. 39) tiene el siguiente comportamiento:

- Si el usuario deja por defecto las dos opciones elegidas la herramienta muestra los 4 pasos (selección de etapas del proyecto, priorización de métodos, revisión de potenciales beneficios y riesgos de negocio y el plan).
- Si el usuario deja seleccionada únicamente la primera opción, la pestaña del paso de revisión de potenciales beneficios y riesgos de negocio desaparece.
- Si el usuario deja seleccionada únicamente la segunda opción, la pestaña del paso de priorización de métodos desaparece.

- Y finalmente, si el usuario no selecciona ninguna opción, se mostraría un mensaje advirtiéndole que, para el uso de la herramienta, al menos una opción tiene que ser elegida.

En cuanto al contenido de la pantalla inicial (ver Fig. 39) en comparación con el *wireframe* (ver Fig. 30) fue simplificado con el fin de mejorar la comunicación.

Fig. 40. Pantalla del paso 1, selección de etapas del ciclo de vida.

El paso 1 (ver Fig. 40) también se simplificó. Se optó por eliminar la segunda columna con el listado de actividades del proceso DCU.

The screenshot shows the 'usability planner' web application. At the top, there is a navigation bar with tabs: Home, Project stages, Methods, Business benefits, Plan, and About. Below the navigation bar, a blue banner reads 'Adjust constraints to obtain recommended methods for each stage considered'. The main content area is divided into three columns:

- Stages considered:** A list with two items: '1. Envisioning opportunities' (highlighted in orange) and '2. System scoping'.
- Constraints:** A section titled 'Stage selected: Envisioning opportunities' with the prompt 'Which constraints below are you agree?'. It contains three sub-sections:
  - Project characteristics: A list of five items with checkboxes. 'Need quick results' and 'Uncertain specification' are checked. The others are 'Very restricted budget', 'Usability important', and 'Rapid feedback required'.
  - User issues: A list of four items with checkboxes, all of which are unchecked: 'Difficult to involve users', 'No access to users', 'Some users have disabilities', and 'Mostly first time users'.
  - Task characteristics: A single item 'Complex task' with an unchecked checkbox.
- Methods:** A section titled 'Recommendations for Envisioning opportunities' with the prompt 'Click on a method to see the weighting for this constraints'. It lists six methods, each preceded by a star rating (three stars):
  - [Future workshop](#)
  - [Preliminary field visit](#)
  - [Focus groups](#)
  - [Photo surveys](#)
  - [Simulation of future use environments](#)
  - [In-depth analysis of work and lifestyles](#)
 At the bottom of this section is a checkbox labeled 'Show methods not recommended'.

Fig. 41. Pantalla del paso 2. Selección de restricciones del proyecto.

En el paso de selección de las restricciones, todos los participantes del test de usuario coincidían en que tener un selector triple para elegir cada una era complejo de entender, la mayoría sugirió tener únicamente dos estados (verdadero o falso) y así se reflejó en la implementación de este paso (ver Fig. 41).



usability planner

Home Project stages Methods **Business benefits** Plan About

How much using usability methods in the selected stage increase potential business benefits or reduce potential risks

**Stages considered**

- 1. Envisioning opportunities
- 2. System scoping

**Potential business benefits**

Stage selected: Envisioning opportunities

- ☐ Reduced development costs
- ☐ Web site usability: improved web sales
- ☐ Product usability: improved product sales
- ☐ Improved productivity: benefits to purchasing organisation
- ☐ Reduced support and maintenance costs

View potential business benefits as a table

**Potential risks**

Stage selected: Envisioning opportunities

- ☐ Increased development costs to produce an acceptable system
- ☐ Web site usability: poor web sales
- ☐ Product usability: poor product sales
- ☐ Poor productivity: risks to purchasing organisation
- ☐ Increased support and maintenance costs

View potential business risks as a table

Modify activities selected

Back Next

About us | Follow the project | Contact | License | Help

2010 Usability planner - Some rights reserved

Fig. 42. Pantalla del paso 3, beneficios de negocio.

En la solución propuesta en el *wireframe*, la parte de negocio estaba presente en los pasos A y C. Si el usuario no tenía interés en planificar métodos de usabilidad teniendo en cuenta variables de negocio (y no lo indicó en la primera pantalla) se encontraba con que, en medio de su flujo de interacción, se mezclaban aspectos de este tipo.

Con el fin de recopilar en una toda la parte relacionada con la planificación en términos de negocio se decidió relegarla al paso 3 en una única vista. De esta forma no rompería el flujo de priorización de métodos.

Por último en la Fig. 43 se puede ver la pantalla en la que el usuario puede seleccionar los potenciales riesgos de negocio para las etapas seleccionadas.

En la Fig. 44 se muestra la pantalla del último paso, el resumen del plan de usabilidad.

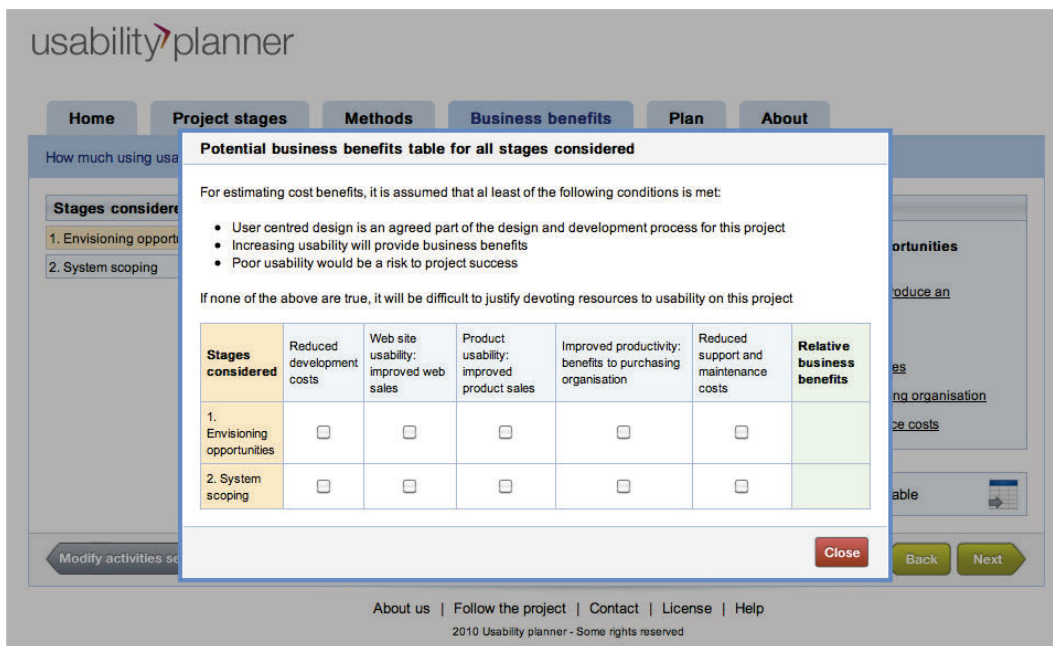


Fig. 43. Tabla con los potenciales riesgos de negocio para las etapas consideradas.

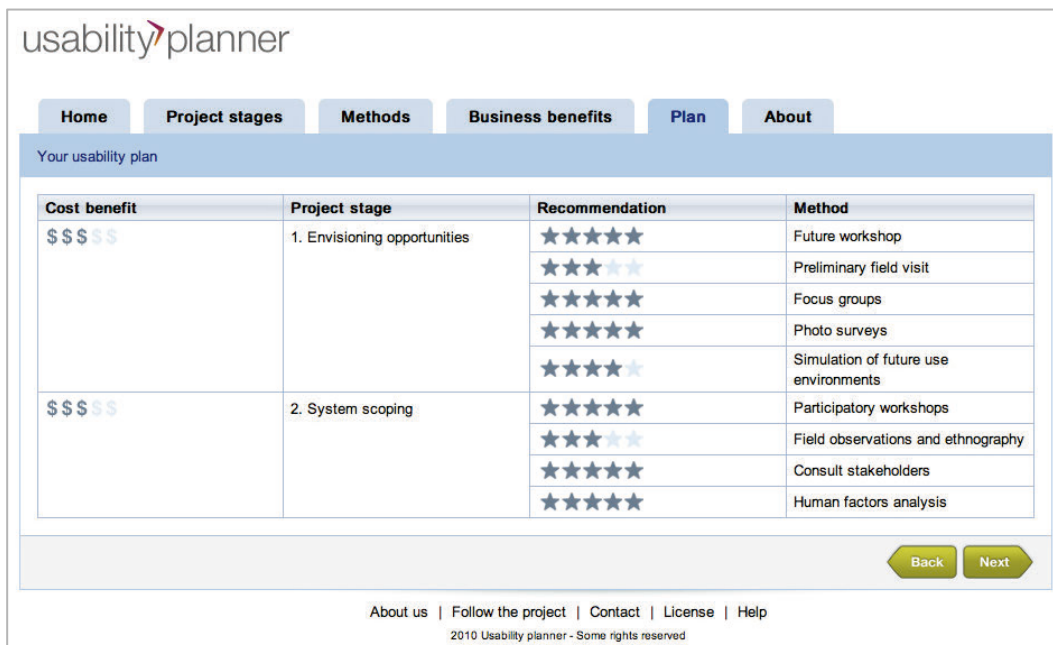


Fig. 44. Pantalla del cuarto paso, resumen del plan.

### 3.3 Tercera iteración.

Tras el desarrollo del segundo prototipo se realizó, nuevamente, una evaluación experta por parte de Bevan y Ferré. En esta evaluación no se definieron grandes cambios, únicamente detalles, que serán expuestos en este apartado.

Lo realmente destacable de esta iteración fue la adecuación del prototipo para su presentación en la 6ª Conferencia Nórdica de Interacción Hombre-Máquina (NordiCHI 2010) que tuvo lugar en Reikiavik, junto con la preparación de un artículo (Antón, Bevan, & Ferré, 2010) y un póster para su demostración.

A nivel conceptual se simplificó notablemente el paso 3 (revisión de potenciales beneficios y riesgos de negocio) quedando únicamente el listado de potenciales beneficios de negocio y potenciales riesgos (ver Fig. 45).

Potential business benefits	Potential risk reduction
<input type="checkbox"/> <u>Reduced development costs</u>	<input type="checkbox"/> <u>Increased development costs</u>
<input type="checkbox"/> <u>Web site usability: improved usage/sales</u>	<input type="checkbox"/> <u>Web site usability: reduced usage/sales</u>
<input type="checkbox"/> <u>Product usability: improved product sales</u>	<input type="checkbox"/> <u>Product usability: reduced product sales</u>
<input type="checkbox"/> <u>Improved productivity: benefits to purchasing organisation</u>	<input type="checkbox"/> <u>Reduced productivity: impact on purchasing organisation</u>
<input type="checkbox"/> <u>Reduced support and maintenance</u>	<input type="checkbox"/> <u>Increased support and maintenance costs</u>

Fig. 45. *Wireframe* de la selección de restricciones de negocio del paso 3.

También se mejoró la interacción del usuario con la selección de las restricciones de proyecto. Se identificó que los usuarios, hasta ese momento, tenían que ir etapa a etapa seleccionando las restricciones, pero nunca tenían una visión de conjunto de todas las etapas con todas las restricciones seleccionadas. Por este motivo se incorporó una vista adicional en el paso 2, en forma de matriz, desde la que poder realizar la selección de forma conjunta.

Overview	1. Concept	1.1 Envisioning opportunities	1.2 System scoping	
Project constraints for each stage	1. Concept	1.1 Envisioning opportunities	1.2 System scoping	Project constraints for each stage
<input checked="" type="checkbox"/> Need quick results	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Need quick results
<input type="checkbox"/> Very restricted budget	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Very restricted budget
<input type="checkbox"/> Usability important	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> Usability important
<input type="checkbox"/> Uncertain specification	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Uncertain specification
User constraints for each stage	1. Concept	1.1 Envisioning opportunities	1.2 System scoping	User constraints for each stage
<input type="checkbox"/> Difficult to involve users	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Difficult to involve users
<input type="checkbox"/> No access to users	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> No access to users

Fig. 46. Selección de restricciones de forma global en el paso 2.

### 3.3.1 Cambios a nivel de diseño gráfico.

Con el objetivo de mejorar el *look & feel* (apariencia, en español) de la interfaz de la aplicación, en esta iteración, se realizó un esfuerzo importante a nivel de diseño gráfico. Debido a que la herramienta iba a ser presentada en un congreso, se consideró importante invertir más tiempo en esta tarea.

El diseño, es una de las piezas clave para transmitir una buena experiencia de usuario. Al fin y al cabo es lo que los usuarios van a ver y sentir. Un diseño inadecuado puede dificultar la interacción con un sistema, puede ocultar lo que debería el usuario ver en un determinado momento o bien puede crear metáforas visuales que lleven al usuario a una mala interpretación de una acción.

En el caso de Usability Planner, se decidió apostar por un diseño minimalista, con colores sólidos y planos, un diseño que no desviase el foco de atención del usuario, que no introdujese metáforas nuevas y que resaltase el carácter guiado de la aplicación.

Por ejemplo, las pestañas que representan cada paso en el prototipo anterior, fueron sustituidas por botones con una flecha de dirección intercalada, para mostrar el carácter secuencial del flujo o paso a paso (ver Fig. 47). La flecha de dirección, además, tenía dos estados, uno gris para indicar que el siguiente paso no estaría disponible hasta que se complete el anterior y uno verde para indicar que se puede avanzar al siguiente paso (o ver qué pasos se han completado).

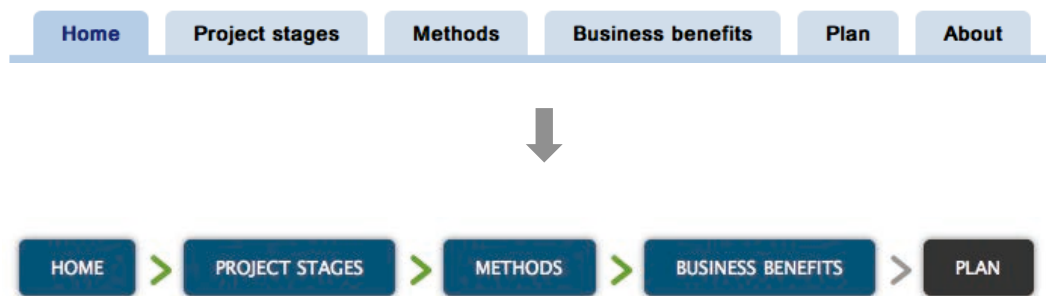


Fig. 47. Cambio de pestañas a botones para el paso a paso.

Se decidió suprimir, del flujo de pasos, la pestaña de *about* (acerca de, en español) ya que no tenía la misma relación que las pestañas / botones anteriores.

Otro detalle creativo que se vio modificado fue el logotipo. Se pensó en seguir la línea minimalista y utilizar únicamente una tipografía. Tras varios bocetos se aprobó el logotipo de la Fig. 48.



Fig. 48. Segundo diseño del logotipo de la aplicación.

### 3.3.2 Ajustes en el prototipo.

En la etapa de desarrollo de esta iteración los cambios más significativos se realizaron en la capa de *front-end* (capa de interfaz de usuario, en español) a nivel de diseño.

También se resolvieron algunos errores funcionales identificados en las pruebas de la iteración anterior y se actualizó la base de datos con todas las técnicas y restricciones que hasta ese momento estaban disponibles.

A continuación se muestra una captura de las pantallas más relevantes de esta versión.

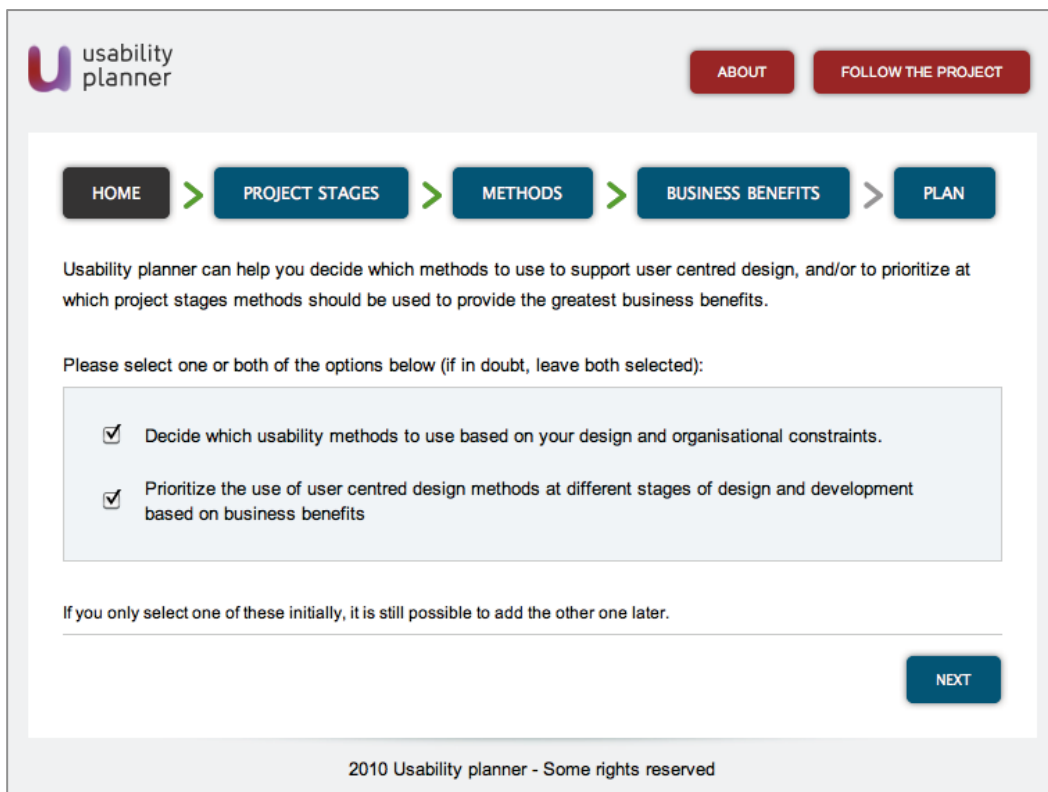


Fig. 49. Pantalla de inicio de la aplicación.

En la pantalla de inicio (ver Fig. 49) se mejoró la redacción de los textos y se añadieron dos botones en la parte superior derecha para ir a dos secciones: “Acerca de esta aplicación” y “Seguir el proyecto”.

usability planner

ABOUT FOLLOW THE PROJECT

HOME > PROJECT STAGES > METHODS > BUSINESS BENEFITS > PLAN

For which stage(s) of your project do you want to select methods or prioritise cost benefits?

☒ Expand all

- ☐ 1. Envisioning opportunities
- ☐ 2. System scoping
- ☒ 3. Understanding needs
  - ☐ 3.1 Context of use
  - ☐ 3.2 Tasks
  - ☐ 3.3 Usability needs
  - ☐ 3.4 Design options
- ☒ 4. Requirements
  - ☐ 4.1 Context requirements
  - ☐ 4.2 Infrastructure requirements
  - ☐ 4.3 User requirements

Fig. 50. Pantalla del paso 2, selección de etapas del proyecto.

usability planner

ABOUT FOLLOW THE PROJECT

HOME > PROJECT STAGES > METHODS > BUSINESS BENEFITS > PLAN

Adjust constraints to obtain recommended methods for each stage considered

Overview 1. Concept 1.1 Envisioning opportunities 1.2 System scoping

Project constraints for each stage	1. Concept	1.1 Envisioning opportunities	1.2 System scoping	Project constraints for each stage
<input checked="" type="checkbox"/> Need quick results	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Need quick results
<input type="checkbox"/> Very restricted budget	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Very restricted budget
<input type="checkbox"/> Usability important	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> Usability important
<input type="checkbox"/> Uncertain specification	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Uncertain specification
User constraints for each stage	1. Concept	1.1 Envisioning opportunities	1.2 System scoping	User constraints for each stage
<input type="checkbox"/> Difficult to involve users	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Difficult to involve users
<input type="checkbox"/> No access to users	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> No access to users

Fig. 51. Pantalla del segundo paso, priorización de métodos (vista general).

usability planner

ABOUT FOLLOW THE PROJECT

HOME > PROJECT STAGES > METHODS > BUSINESS BENEFITS > PLAN

Adjust constraints to obtain recommended methods for each stage considered

Overview 1. Concept 1.1 Envisioning opportunities 1.2 System scoping

**Project constraints**

- ☒ Need quick results
- ☒ Very restricted budget
- ☐ Usability important
- ☐ Uncertain specification

**User constraints**

- ☐ Difficult to involve users
- ☐ No access to users
- ☐ Some users have disabilities

**Methods and Techniques**

- Preliminary field visit (2)
- Photo surveys (2)
- In-depth analysis of work and lifestyles (0)
- Focus groups (-1)
- Future workshop (-2)
- Simulations of future working environments (-2)

Fig. 52. Pantalla del segundo paso, priorización de métodos (vista por actividad).

usability planner

ABOUT FOLLOW THE PROJECT

HOME > PROJECT STAGES > METHODS > BUSINESS BENEFITS > PLAN

For each stage of design and development, indicate whether using usability methods would significantly increase potential cost benefits, or significantly reduce project risks.

Increase potential cost benefits Reduce potential risks


Cost benefits for each stage	1. Concept	1.1 Envisioning opportunities	1.2 System scoping	Cost benefits for each stage
<input checked="" type="checkbox"/> Development: reduced time/costs	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Development: reduced time/costs
<input type="checkbox"/> Product usability: improved product sales	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Product usability: improved product sales
<input type="checkbox"/> Web site usability: improved usage/sales	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Web site usability: improved usage/sales
<input type="checkbox"/> Efficiency in use: increased productivity	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Efficiency in use: increased productivity
<input type="checkbox"/> Support and maintenance: reduced costs	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> Support and maintenance: reduced costs

Fig. 53. Pantalla del tercer paso, beneficios de negocio.



### 3.3.2 Diseño del póster y creación de un artículo.

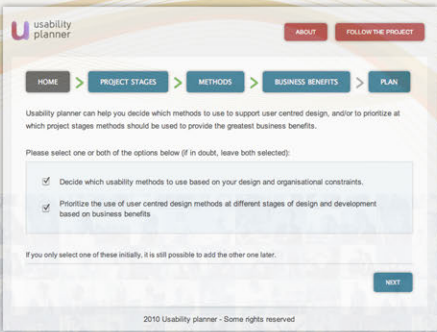
Con el fin de promocionar la herramienta en el NordiCHI y como reclamo para que profesionales del mundo de la usabilidad probaran la aplicación y nos dieran su opinión y se elaboró el póster de la Fig. 54 y un artículo (Antón, Bevan, & Ferré, 2010) para dar a conocer el proyecto y la herramienta.



www.usabilityplanner.org

*Which of the potential methods that could be used to achieve each activity would be most appropriate?*

*Which UCD activities would provide the greatest cost-benefits or risk mitigation?*



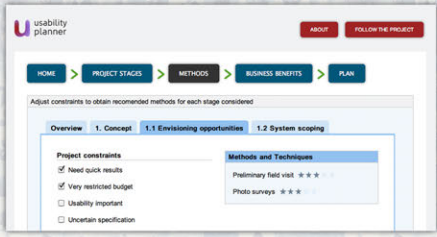
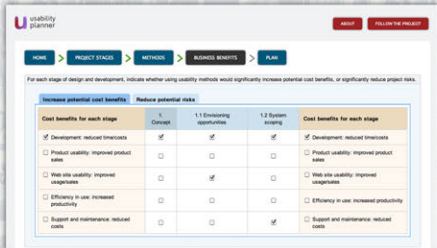

Public domain tool  
Configurable to apply additional sources of expertise in selection of usability methods.

## UCD Method Selection with Usability Planner

*Try the tool, and give us your feedback!*

For the project stages chosen:

- ➔ According to the specific project constraints, most appropriate usability methods are suggested for each stage.
- ➔ Estimation of cost-benefit issues is facilitated by providing a list of possible business benefits associated with the application of usability practices, and potential risks of not doing so.
- ➔ A plan gathers all UCD methods per project stage, with the addition of business considerations when applicable.






Select project stages

Adjust constraints for each stage

Select business benefits or risks

Plan



Nigel Bevan - Xavier Ferre - Tomás Antón Escobar

Fig. 54. Póster presentado en el NordiCHI 2010.

### 3.3.3 Evaluación del prototipo durante el congreso.

Durante el transcurso del NordiCHI se tuvo la oportunidad de mostrar la aplicación a un gran número de profesionales en experiencia de usuario. Se dispuso un Mac sobre un atril junto con el póster de forma que, aquellas personas que se interesaron en conocer la herramienta, pudieran probarla in-situ.

Previamente, se acordaron las siguientes tareas (las mismas que se utilizaron en el test de usuario de la iteración anterior):

1. “¿Qué técnicas de usabilidad emplearías en tu proyecto en la sub-etapa de análisis del contexto de uso? ¿Qué buenas prácticas se recomiendan para su aplicación?”
2. “¿Qué técnicas de usabilidad utilizarías en el caso de que tu presupuesto sea ajustado y tengas un difícil acceso a los usuarios para la realización del proyecto?”

Y se añadieron nuevas preguntas (en el tercer anexo se incluye la plantilla del cuestionario completo que fue entregado a cada participante):

1. ¿Cómo mejorarías esta aplicación?
2. ¿Te imaginas usando esta aplicación? ¿Para que la usarías?
3. ¿Imaginas a alguna persona de tu equipo utilizando esta aplicación?

Estas fueron las respuestas a las 3 preguntas anteriores más destacables de los participantes:

Usuario 1	
<b>Ocupación</b>	Investigador en la Katholieke Universiteit Leuven.
<b>Pregunta 1</b>	<ul style="list-style-type: none"><li>- Cómo añadir nuevas restricciones, ya que si la herramienta no te ofrece una restricción cercana a tus tareas probablemente no elijas la adecuada.</li><li>- La restricción: “Propósito general: Usado en distintos contextos” no se entiende.</li><li>- No se entiende la finalidad del paso “Beneficios de negocio”. No se entiende que ese paso es para decidir que etapas del ciclo de vida proveen más beneficio... ¿Si sólo tengo elegida una etapa? No tendría mucho sentido.</li></ul>

<b>Pregunta 2</b>	Sí, perfectamente.
<b>Pregunta 3</b>	Imagino a desarrolladores con interés en usabilidad.
<b>Comentarios</b>	<ul style="list-style-type: none"> <li>- El paso de beneficios de negocio es para expertos en usabilidad.</li> <li>- Me perdí al pulsar en el botón “Siguiente” en el paso de priorización de métodos. Pensé que ese botón me mostraría más restricciones.</li> <li>- ¿Por qué algunos métodos aparecen con estrellas y otros no?</li> <li>- ¿Qué significado tienen las estrellas? ¿Están cuantificadas?</li> <li>- Me he dado cuenta que la aplicación no da soporte para proyectos dirigidos a la creación de aplicaciones móviles.</li> </ul>

<b>Usuario 2</b>	
<b>Ocupación</b>	Profesor de universidad
<b>Pregunta 1</b>	Mejoraría el uso del botón de avance y retroceso por los pasos de la aplicación.
<b>Pregunta 2</b>	Sí, puede ayudar a la enseñanza.

<b>Usuario 3</b>	
<b>Ocupación</b>	Profesor en la Universidad de Dinamarca
<b>Comentarios</b>	No acabo de entender el número de estrellas. ¿Hay una escala? ¿Cuál es el número máximo de estrellas?

<b>Usuario 4</b>	
<b>Ocupación</b>	Investigadora en Nokia
<b>Comentarios</b>	Me gustaría que la herramienta, al lado de cada etapa, tuviera un icono con una interrogación “?” para mostrar información adicional.

<b>Usuario 5</b>	
<b>Ocupación</b>	Profesor de universidad
<b>Pregunta 3</b>	Lo veo muy útil para mis alumnos. Ellos hacen diseño industrial y realizan prototipos que prueban y mejoran iterativamente.

<b>Comentarios</b>	No comprendo el nombre de las etapas del ciclo de vida elegidas.
--------------------	--

<b>Usuario 6</b>	
<b>Ocupación</b>	Estudiante de doctorado en interfaces multimodales y gestuales.
<b>Comentarios</b>	<ul style="list-style-type: none"> <li>- No mostraría sub-etapas dentro de cada etapa.</li> <li>- El botón de siguiente desaparece cuando la vista se expande y aparece el <i>scroll</i>, estaría mejor si el botón estuviese siempre presente en la ventana.</li> <li>- Me resultó confuso, una vez seleccionadas las restricciones, ver un botón de siguiente.</li> <li>- No entiendo el uso del paso “Beneficios de negocio”, qué aporta, es confuso. De hecho, ya tengo lo que quería, las técnicas, probablemente cerraría en este punto la herramienta.</li> <li>- Hay demasiados métodos en la lista, yo solo mostraría los 3 mejores.</li> </ul>

### 3.4 Cuarta iteración.

La presentación de la aplicación en Reikiavik, ante una audiencia tan especializada y a la vez multidisciplinar, fue una gran oportunidad para comprobar si realmente el grado de usabilidad alcanzado era el esperado. Se tuvo la oportunidad de entrevistar a potenciales usuarios y de recoger su opinión.

La impresión general fue que la herramienta iba por buen camino, que realmente respondía a las expectativas y necesidades de los usuarios, pero aún necesitaba algunos ajustes para mejorar su nivel de usabilidad.

Esta iteración destaca por la inclusión de una nueva Persona, un nuevo escenario, el desarrollo de un nuevo prototipo y su presentación en Vancouver (Canadá) en el Congreso de Interacción Hombre – Máquina de 2011 (CHI 2011).

### 3.4.1 Inclusión de una nueva Persona.

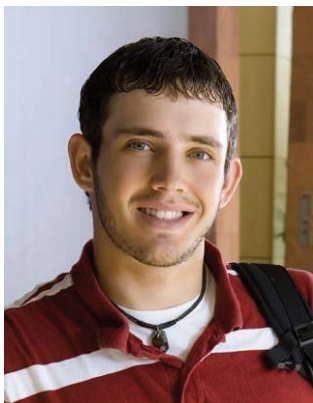
Debido a que en el NordiCHI muchos estudiantes y desarrolladores mostraron gran interés por la aplicación se decidió introducir una nueva persona. Esto tenía grandes implicaciones en el estado actual de la herramienta, ya que hasta ahora el enfoque había sido principalmente centrado en profesionales de la usabilidad.

El nuevo perfil de usuario a incluir no cuenta con conocimientos de usabilidad pero tiene interés en aprender sobre el tema y aplicar métodos y técnicas en función de las limitaciones del proyecto. Escenarios en los que el presupuesto del proyecto es ajustado o no hay acceso a usuarios para realizar pruebas de usabilidad son las limitaciones para las cuales la aplicación puede ser de gran ayuda.



**Jaime** / 24 años / Español / Desarrollador

---



*Jaime lleva un año trabajando como desarrollador de aplicaciones web y móviles para la empresa HealthSoft.*

*Jaime ha leído, en numerosas ocasiones, las ventajas de llevar a cabo un proyecto desde el enfoque DCU y aunque, por su posición actual, no está capacitado para mejorar los procesos internos de su empresa, está muy interesado en saber aplicar métodos y técnicas de usabilidad para mejorar la calidad de los sistemas que desarrolla.*

*Jaime tiene dos problemas recurrentes en sus proyectos: la falta de tiempo y la limitación de presupuesto. Normalmente, sus jefes de proyecto, le asignan menos de la mitad de horas que necesita para llevar a cabo un desarrollo de calidad.*

*A Jaime le gustaría tener una aplicación que le ayudase a identificar las técnicas de usabilidad que minimicen el tiempo de desarrollo y mejoren la calidad final del resultado.*

*A Jaime le encantan los deportes de aventura y el surf. Le encantan las series de TV americanas. En un futuro se ve dirigiendo un equipo de desarrolladores.*

El primer problema que se identificó, al introducir este nuevo perfil, fue que las etapas del ciclo de vida que maneja la herramienta (basadas en el ISO PAS 18152) no serían familiares para desarrolladores o estudiantes de Ingeniería Informática, por lo que se decidió incluir la opción de poder elegir las etapas del ciclo de vida del Proceso Unificado.

Este cambio implicó modificar toda la lógica de la aplicación, ya que los métodos se priorizaban en base a las etapas definidas en el ISO PAS 18152 y al introducir un nuevo repertorio de etapas (las del Proceso Unificado) hubo que ajustar y crear nuevas reglas.

### 3.4.2 Creación de un segundo escenario.

Con el fin de comprender cómo Jaime interactuaría con la aplicación se creó un nuevo escenario de uso:

*Jaime es desarrollador de software en HeathSoft, una empresa dedicada al desarrollo de software médico. Se encuentra realizando un proyecto llamado HDB que permite registrar los historiales de los pacientes en función de un nuevo estándar.*

*El proyecto se encuentra en fase de desarrollo en este momento. El departamento financiero ha comunicado que el presupuesto se está agotando por lo que Jaime se ve en la necesidad de reducir los costes inmediatamente y optimizar recursos. Por otro lado, Jaime está teniendo muchas dificultades para contactar con los doctores para obtener soporte y comentarios.*

*Él necesita priorizar las actividades del proceso DCU que mejor se ajustan al escenario en el que se encuentra y para ello utiliza Usability Planner.*

*Ya en la herramienta, primero indica que quiere utilizar las actividades del proceso unificado, ya que encajan con las etapas de su proyecto. A continuación selecciona la etapa de desarrollo. Marca las siguientes restricciones asociadas al proyecto:*

- ☒ *Se requieren resultados rápidos.*
- ☒ *Presupuesto muy ajustado.*

*Y las siguientes relacionadas con los usuarios:*

- ☒ *Es difícil involucrar a los usuarios.*

*Jaime, según marca las restricciones observa como el ranking de métodos va variando en función de su selección.*

*Finalmente, Usability Planner le recomienda emplear la técnica de Usability walkthrough.*

### 3.4.3 Revisión de las tareas.

En la siguiente tabla de tareas se han incluido aquellas que responden a las metas y objetivos de Jaime. Además se incluyó una nueva tarea, que fue sugerida por algunos participantes durante las pruebas de la iteración anterior, consistente en permitir a los usuarios eliminar del ranking aquellas técnicas que por cualquier motivo no apliquen en el contexto de su proyecto u organización.

Tabla 7. Tareas asociadas a las personas identificadas.

	Tarea	Sara	Carmen	Steven	Jaime
1	Seleccionar actividades DCU del ISO PAS 18152 y consultar las buenas prácticas.	□	□	□	×
2	Seleccionar los beneficios o riesgos en términos de negocio que consideren en cada etapa.	×	×	□	×
3	Indicar las limitaciones que su proyecto tiene en función de las etapas o subetapas elegidas. Por ejemplo falta de presupuesto.	○	□	□	□
4	Ver priorizadas las técnicas y métodos que la herramienta recomienda.	□	□	□	□
5	Adaptar la aplicación en base a su contexto organizacional para afinar la lista de técnicas y métodos recomendados.	×	×	□	×
6	Seleccionar actividades del Proceso Unificado y consultar las buenas prácticas asociadas.	×	×	○	□
7	Eliminar técnicas que no aplican en el contexto del proyecto o de la organización.	○	□	□	○
□ Tarea asociada al usuario, × Tarea no asociada el usuario, ○ Neutro.					



### 3.4.4 Producción de una nueva solución de diseño.

Al igual que en las iteraciones anteriores, con los resultados de las evaluaciones y la inclusión del nueva Persona, junto con sus tareas, se realizó un nuevo *wireframe* que recogiera las modificaciones de la aplicación. El prototipo en papel fue evaluado y aprobado por Bevan y Ferré.

Se llevó a cabo la implementación de las nuevas funcionalidades junto con un rediseño del *look & feel* que contribuyera a mejorar la comprensión y localización de los contenidos de la aplicación.

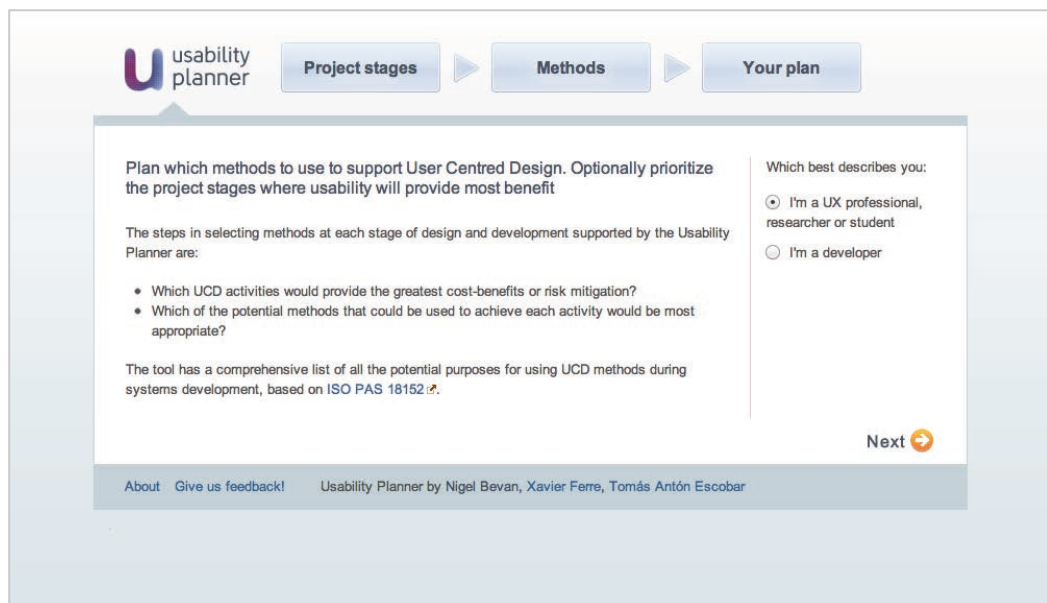


Fig. 55. Pantalla de inicio de la aplicación.

Como se puede observar en la Fig. 55, en la parte derecha del contenido principal, se ha añadido un selector para que el usuario elija el perfil que mejor le describa.

usability planner

Project stages Methods Your plan

Stages selection Prioritize stages

For which project stage(s) do you want to plan methods to introduce usability improvements into your project?  
Select project stage(s) to plan the most appropriate methods to achieve the best practices. Optionally prioritize the stages where usability will provide most benefit

Expand all Collapse all

- ☐ 1. Inception  
more information
- ☐ 2. Elaboration  
more information
- ☐ 3. Construction  
more information
- ☐ 4. Deployment  
more information

Back Next

About Give us feedback! Usability Planner by Nigel Bevan, Xavier Ferre, Tomás Antón Escobar

Fig. 56. Selección de actividades del Proceso Unificado.

En la Fig. 56 se puede observar la inclusión de la selección de las etapas del ciclo de vida del Proceso Unificado.

1. Concept 1.1 Envisioning opportunities

<input checked="" type="checkbox"/> Braindrawing	★★★★☆
<input checked="" type="checkbox"/> Brainstorming	★★★★☆
<input checked="" type="checkbox"/> Preliminary field visit	★★★★☆
<input checked="" type="checkbox"/> Photo study	★★★☆☆
<input checked="" type="checkbox"/> Focus groups	☆☆☆☆☆
<input checked="" type="checkbox"/> Simulations of future working environments	☆☆☆☆☆

Fig. 57. Listado de técnicas y métodos con opción de eliminar.

En la Fig. 57 se puede ver cómo se cuantificó el número de estrellas que cada método podría alcanzar en el ranking.

Además se incluyó la opción de eliminar del ranking las técnicas y métodos que el usuario considerase que no aplican en su contexto u organización. Se realizó incluyendo un icono de eliminar al lado del nombre de cada método o técnica tanto en el ranking final como en la selección de restricciones.

### 3.4.5 Evaluación de la versión alcanzada.

El congreso de Interacción Hombre – Máquina de 2011 (CHI 2011) en Vancouver (Canadá) sirvió de escenario para dar a conocer y recoger la opinión de profesionales ligados a la experiencia de usuario. La mayoría no tuvieron dificultades significativas para resolver las dos tareas que se les plantearon (las mismas que en evaluaciones anteriores) y la mayoría destacó la facilidad de uso y la utilidad de la aplicación.

Los comentarios e impresiones recibidos en Vancouver sirvieron para confirmar que la aplicación cumplía con las necesidades de los usuarios ligados profesionalmente a la experiencia de usuario.

No obstante, ya que los cambios más significativos que se realizaron en la aplicación eran los relacionados con la introducción de un nuevo perfil de usuario (estudiantes y profesionales ligados al desarrollo de software) se decidió realizar una evaluación con un público objetivo más ligado a este ámbito.

Se pidió la colaboración de 14 alumnos de la asignatura de Diseño de la Interacción, cursada entre febrero y junio de 2011, en el *European Master on Software Engineering*.

Estas fueron las preguntas y las respuestas más destacadas de la encuesta que se les pidió completar tras el uso de la aplicación para resolver un trabajo práctico consistente en seleccionar los métodos de usabilidad más adecuados para resolver con éxito un determinado proyecto.

1. Errores cometidos por una interpretación incorrecta de los textos de la aplicación (contenido, etiquetas, botones...):
  - a. Comentarios de la pantalla de selección de restricciones:
    - *Restricción “un producto bien definido”: Es importante que los usuarios comprueben bien como se utiliza el producto sin necesidad de ningún tipo de formación.*

b. Comentarios acerca del ranking (estrellas):

Un estudiante entendió que el número de estrellas significa cuánto de aplicable es una determinada técnica (en porcentaje) en el proyecto.

Otro estudiante al ver solo dos estrellas en una técnica interpretó que esa técnica era poco recomendada.

También se comentó que Usability Planner recomienda la técnica “Diseño participativo” como muy recomendada a pesar de que la experiencia del usuario en usabilidad sea escasa.

2. Comentarios sobre algunas técnicas y métodos de usabilidad que la aplicación recomienda y que los participantes indican que de forma errónea:

- *Nosotros hemos estudiado que el éxito de un método depende, en gran medida, de la experiencia del profesional que lo ponga en práctica, sin embargo, Usability Planner sugiere aplicar ciertos métodos complejos aunque la restricción – No existe experiencia en usabilidad – esté marcada.*

- *Usability Planner recomienda tormenta de dibujos en lugar de tormenta de ideas y considero que ambas tienen el mismo efecto y resultados.*

3. Comentarios sobre métodos que Usability Planner no incluye y debería:

- *Nosotros usamos recorrido cognitivo y evaluaciones heurísticas pero en la aplicación no se recomiendan.*

- *Nosotros usamos dos tipos de encuestas: De satisfacción y tras un test de usuario. Usability Planner no los recomienda, a pesar de que son dos técnicas que proveen información muy útil.*

4. Comentarios sobre métodos que la aplicación recomienda y que los participantes indican que de forma apropiada:

- *Aquí puedo ver como la aplicación me recomienda que utilice sobre todo la técnica de tormenta de ideas y efectivamente coincido en que es la que se debería usar.*

- *Paseo cognitivo aparece como poco recomendada y efectivamente coincide en que no es el método más adecuado ya que requiere de experiencia avanzada. La aplicación confirma mi decisión.*

5. Mejoras que sugirieron para futuras versiones:

- *Quizá los métodos y técnicas de usabilidad deberían mostrarse al usuario de la misma forma que se muestran los productos en una web de comercio electrónico.*

- *Capacidad para guardar el plan de usabilidad y poderlo recuperar y editarlo.*

- *Capacidad de compartir el plan con otros profesionales, por ejemplo un compañero de trabajo.*

- *Se debería poder mostrar los métodos junto con sus descripciones.*

6. Sugerencias para mejorar el criterio de selección de métodos:

- *Muchos métodos no tienen disponible información adicional o una descripción.*

- *La aplicación no estima el tiempo que lleva la aplicación de un determinado método.*

7. Comentarios acerca de la utilidad de la aplicación:

- *Algunas técnicas la mayoría de las veces son beneficiosas y deberían aplicarse independientemente de las características del proyecto. Por ejemplo: escenarios de uso, tormenta de ideas y evaluación heurística. No obstante, la aplicación indica métodos menos conocidos que son más específicos para determinados proyectos.*

- *En mi opinión, la aplicación puede ser usada como referencia para seleccionar los métodos más apropiados para un determinado proyecto, no obstante no puedes seguir al 100 % sus recomendaciones.*

Tras la evaluación se corrigieron algunos fallos a nivel funcional que hacían que la lista priorizada de métodos no fuera del todo correcta y se afinó el motor de inferencia para mejorar los resultados del ranking.

Otras mejoras, indicadas por los usuarios, se analizaron y se consideró que no era necesario su inclusión para lanzar la primera versión estable de la aplicación.

Finalmente, tras cuatro iteraciones la versión 1.0 de Usability Planner se lanzó oficialmente.

## Capítulo 4.

# ESTUDIO DE TECNOLOGÍAS WEB PARA LA IMPLEMENTACIÓN.

Tal como se vio en el capítulo anterior, Usability Planner fue inicialmente concebido y diseñado para ser una aplicación de escritorio. Como solución tecnológica para su implementación se eligió Java por su carácter multiplataforma y su alta mantenibilidad.

Después de evaluar el resultado, se planteó dar el salto a la web por sus ventajas, como la capacidad de difusión y la disponibilidad. De ahí que el reto fuera buscar aquellas tecnologías que no comprometieran los requisitos del proyecto y mantuvieran la mayor parte del desarrollo, orientado a objetos, ya realizado en Java para la versión de escritorio.

Este capítulo resume un estudio realizado, al comienzo de la segunda iteración del proceso de diseño de la aplicación, que recorre las tecnologías de desarrollo web que se evaluaron como factibles para realizar la transición de escritorio a web y que mejor se adaptan a la naturaleza y necesidades del proyecto.

El estudio comienza con un recorrido por los distintos tipos de aplicaciones que a día de hoy conviven en la web, así como las tecnologías que las implementan. David Siegel establece 3 generaciones de sitios web (Siegel, 1996) pero, a día de escribir este documento, se podría, incluso añadir una cuarta generación basada fundamentalmente en la interoperabilidad, la colaboración, la creación dinámica de contenidos y la experiencia de usuario.

## 4.1 Aplicaciones web de primera y segunda generación.

En noviembre de 1990 Tim Berners-Lee, empleado del CERN (Consejo Europeo para la Investigación Nuclear), puso en funcionamiento el primer servidor web y publicó la primera página web. No obstante, no fue hasta 1993 cuando Mosaic (ver Fig. 58), un navegador web gráfico desarrollado en el Centro Nacional para Aplicaciones de Supercómputo (o NCSA por sus siglas en inglés), se convirtió en la ventana gratuita de acceso a servicios de noticias, servicios educativos interactivos, catálogos, informes de tráfico, etc.

Siguiendo la clasificación de David Siegel (Siegel, 1996) la primera generación de páginas web, que abarca desde 1992 hasta 1994, básicamente ofrecían información textual estática al usuario. Se utilizaba el lenguaje de marcado HTML (*HyperText Markup Language*, en español Lenguaje de Marcado de Hipertexto), el cual era el responsable de la estructura y contenido de cada página. Las primeras páginas hacían uso exclusivamente de esta tecnología para hacer llegar al usuario un determinado servicio.

La web de segunda generación nació a partir de 1995 con la inclusión de elementos gráficos y la aparición de los *applets* de Java. Estos permitieron que las páginas tuvieran pequeños programas que se ejecutaban en el contexto del navegador, posibilitando nuevas funcionalidades.



Fig. 58 Mosaic 1.0 funcionando en un Mac System 7.1



## 4.2 Aplicaciones web de tercera generación.

La llegada a finales del año 1995 del lenguaje JavaScript y en 1996 de las hojas de estilo en cascada (o CSS) definieron la llamada tercera generación de aplicaciones web.

JavaScript de Netscape, es un lenguaje de script pensado para ser usado en aplicaciones web. Su éxito radica en el manejo del Modelo de Objetos de Documento (o DOM por sus siglas inglés), es decir, la manipulación de los objetos que contiene una página una vez esta ha sido enviada al cliente o navegador.

En el año 1996 la W3C (World Wide Web Consortium), organismo encargado de crear todos los estándares relacionados con la web, publicó la primera recomendación oficial de las hojas de estilo en cascada o CSS (Cascading Style Sheets), mediante las cuales se podía dotar a las etiquetas HTML de estilos y, lo que aún era más importante, poder separar el estilo del contenido de la página.

Ambas tecnologías abrieron las puertas al desarrollo de páginas con nuevas funcionalidades y características que rompían con la experiencia estática que el usuario tenía de la web. Éstas fueron la antesala de una oleada nueva de servicios, de una nueva generación de aplicaciones, donde el usuario pasaría de ser un mero observador a ser el protagonista.

## 4.3 Aplicaciones web de cuarta generación.

Esta generación de aplicaciones está basada en la interoperabilidad, el Diseño Centrado en el Usuario (Sharma, 2011) y la colaboración. Blogs, redes sociales, comercio electrónico o wikis forman la llamada Web 2.0, término que fue acuñado por Dale Dougherty en el año 2004 y más tarde presentado por Tim O'Reilly en la conferencia Web 2.0 que tuvo lugar en San Francisco, California en 2004 (O'Reilly, 2005) haciendo referencia a un antes y un después y a un verdadero salto en la visión que el usuario tenía de la web.

Hasta la llegada de esta nueva generación, la unidad en torno a la cual se estructuraban los contenidos en la web era la página. Pero esta unidad comenzó a quedarse pequeña con la aparición de servicios que distan considerablemente de los que se prestaban en la década del los 90.

Los términos aplicación, herramienta, sistema, siempre ligados al software de escritorio, comenzaron a tomar un protagonismo distinto en un nuevo marco, la web. Y esto se debió, principalmente, a la evolución de las tecnologías que permitieron la implementación de “páginas” como Facebook o Gmail, verdaderas aplicaciones o sistemas embebidos en una web.

#### 4.3.1 Tecnologías que implementan la web 2.0.

La nueva generación de aplicaciones más conocida como la web 2.0 ha ido enriqueciendo la web aportando nuevos servicios que han mejorado la experiencia de usuario y han dado lugar a un nuevo modelo de desarrollo de software y tecnologías.

En este sentido existen numerosas tecnologías, lenguajes y *frameworks* de desarrollo que permiten implementar aplicaciones web 2.0, los más utilizados son:

- **PHP:** Comenzó a utilizarse en la web en 1998. Es un lenguaje interpretado y no compilado, esto permite incluir fragmentos de código en el propio HTML para realizar tareas dinámicas y establecer comunicaciones cliente-servidor. PHP se ha convertido en uno de los lenguajes más utilizados para el desarrollo de aplicaciones web. Funciona muy bien de forma conjunta con MySQL (Davis & Phillips, 2008). Requiere de un servidor, habitualmente Apache, para ser interpretado.
- **JSP** (Java Server Pages): Al igual que PHP permite generar contenido dinámico. Una página escrita en JSP debe ser interpretada por un servidor de aplicaciones (Tomcat, Glasfish...). La ventaja de JSP, con respecto a otros lenguajes, es que se basa en el lenguaje Java, el cual está presente más allá del ámbito web y permite el desarrollo de cualquier aplicación informática (Cadenhead & Lemay, 2007).
- **STRUTS** y **JSF** (Java Server Faces): Ambos facilitan el desarrollo de aplicaciones web bajo el patrón MVC (Model View Controller, en español Modelo Vista Controlador). Los dos están muy ligados a JSP enriqueciéndolo mediante la incorporación de bibliotecas de etiquetas. JSF simplifica el desarrollo de interfaces de usuario mediante la agregación de APIs para representar componentes de la interfaz de usuario, manejar eventos etc. (ORACLE, 2010).

- **ASP:** Es la tecnología de Microsoft para el desarrollo de webs dinámicas. Su primera versión fue lanzada en 1998. Requiere de un servidor IIS (Internet Information Services) también de Microsoft para funcionar. Su filosofía y funcionamiento es similar al de JSP o PHP, provee de objetos para facilitar el desarrollo. En el año 2002 fue sustituido por **ASP.NET**, el cual permitió, a los desarrolladores, implementar aplicaciones web utilizando cualquier lenguaje soportado por la plataforma .NET.
- **Ruby on Rails:** Es un *framework*, de código abierto, basado en el lenguaje de programación Ruby. Su popularidad creció en 2006 debido a su filosofía de escribir el menor código posible para lograr grandes resultados (Tate & Hibbs, 2007). Al igual que el resto de tecnologías necesita un servidor web (por ejemplo Apache) para ser interpretado.
- **AJAX** (Asynchronous JavaScript and XML, en español, JavaScript Asíncrono y XML): Surgió en el año 2005 como la tecnología para el desarrollo de aplicaciones Web dinámicas. Supuso un gran avance al permitir la actualización del contenido de las páginas, sin necesidad de realizar una recarga de las mismas. AJAX abrió la puerta a una nueva familia de aplicaciones, las llamadas **aplicaciones RIA** (*Rich Internet Applications*, o Aplicaciones de Internet Ricas, en español), que se detallan en la siguiente sección.

A las tecnologías anteriores hay que sumar un nuevo lenguaje de marcando, el **HTML5**, actualmente en fase experimental, cuyo primer borrador fue publicado en 2008 y con previsión de publicarse un borrador definitivo para el año 2012 (Sanders, 2011). HTML5 incorpora nuevas etiquetas y elementos para dar soporte a contenidos multimedia, y mejorar la estructura de las páginas dotándolas de significado semántico.

Las aplicaciones web se basan en una arquitectura cliente – servidor, siendo el cliente el navegador del usuario y el servidor la máquina dónde reside la aplicación implementada mediante una tecnología o solución de desarrollo. Existe un tercer elemento que da soporte a los datos que manejan estas aplicaciones, la base de datos. Son numerosas las soluciones para guardar los contenidos que manejan las aplicaciones, aquí se destacan tres:

- **MySQL:** Es un sistema de gestión de bases de datos relacionales, multihilo y multiusuario bajo licencia GPL (para desarrollos de código abierto). Funciona en numerosas plataformas (AIX, FreeBSD, Linux, OSX...). Facebook, Twitter o Google lo utilizan. Su rendimiento es similar a algunas bases de datos comerciales como Oracle y SQLServer (Davis & Phillips, 2008).
- **SQLServer:** Es el sistema de bases de datos relacionales de Microsoft. La versión Express Edition se distribuye gratuitamente.
- **XML:** Siglas en inglés de Extensible Markup Language (lenguaje de marcas extensible en español) desarrollado por la W3C y propuesto como estándar para el intercambio de información estructurada entre distintas plataformas.

#### 4.3.2 Aplicaciones RIA.

Las aplicaciones RIA son webs enriquecidas con características propias de las aplicaciones de escritorio, que mejoran sustancialmente la experiencia del usuario en la web.

Las aplicaciones web, presentadas hasta ahora, están constituidas por páginas que se cargan en el navegador a medida de que el usuario pulsa sobre sus enlaces, estableciéndose una comunicación cliente – servidor. Si la aplicación tuviera que mostrar o actualizar un contenido sería necesaria la recarga completa de la página. Las aplicaciones RIA, sin embargo, se cargan completamente en el navegador la primera vez que el usuario accede a ellas y sólo realizan peticiones al servidor cuando se necesitan datos externos.

Al igual que las aplicaciones de escritorio, las aplicaciones RIA se constituyen en vistas, diluyéndose el concepto de página. Esto posibilita incorporar nuevas características a las aplicaciones web que mejoran la experiencia del usuario, como por ejemplo multimedia, mecanismos de interacción como el *drag and drop* (en español, arrastrar y soltar) o funciones como copiar y pegar.

Otra característica que las diferencia del resto de aplicaciones, pero curiosamente comparte con los *applets* de Java, es que requieren de un *sandbox* o máquina virtual que se instala en el navegador del usuario<sup>5</sup>.

Para ilustrar las capacidades que ofrecen las aplicaciones RIA se va tomar como ejemplo la aplicación *Presentation* de Google (ver Fig. 59). Se trata de un editor de presentaciones en línea que permite crear publicaciones con texto enriquecido, animaciones, transiciones con una amplia variedad de estilos y temas. Podría decirse que es el equivalente a PowerPoint, la popular aplicación de escritorio de Microsoft.

Presentations de Google, al tratarse de una aplicación Web, está disponible en cualquier dispositivo conectado a Internet y los documentos generados pueden revisarse y ser compartidos en línea.

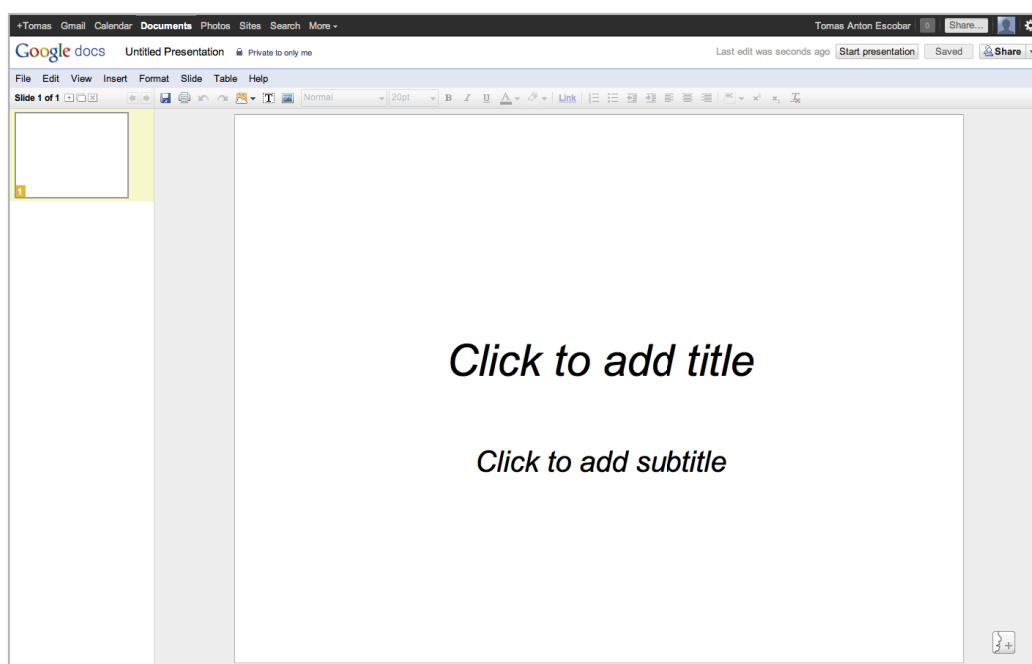


Fig. 59 Interfaz de Presentation.

---

<sup>5</sup> Las aplicaciones RIA implementadas exclusivamente en AJAX, al estar soportado por la mayoría de los navegadores, no requieren de ninguna máquina virtual para funcionar.

Como puede observarse en la Fig. 59 el aspecto de Presentation es muy similar al de las aplicaciones de escritorio homólogas. Desde ella se pueden crear diapositivas tal y cómo se crearían en una aplicación de escritorio, llevando al usuario al punto de olvidar que se encuentra en una página web.

Otras características que distinguen a las aplicaciones RIA del resto de aplicaciones web son:

- Balance cliente - servidor. En las aplicaciones web la lógica de negocio reside completamente en el servidor. En las aplicaciones RIA el modelo puede coexistir tanto en el cliente como en el servidor.
- Eficiente comunicación asíncrona entre cliente y servidor.
- Reducción del tráfico de la red. Las aplicaciones RIA contienen pocos enlaces a otras páginas (incluso pueden contener ninguno) y minimizan por tanto el intercambio de datos con el servidor.
- Las aplicaciones, una vez descargadas en el cliente, pueden no depender de una conexión con el servidor.
- Están soportadas por los sistemas operativos y navegadores más utilizados por los usuarios.
- Implementan funciones muy utilizadas por el usuario en el ámbito del escritorio como copiar, cortar y pegar.
- Más capacidad de respuesta, el usuario no necesita recargar la página para obtener un resultado.

Existen diferentes soluciones para la implementación de aplicaciones RIA, con distintos niveles de penetración en la web:

A continuación se describen las más representativas:

- **AJAX:** Ya se ha presentado anteriormente. El hecho de estar basado en el estándar JavaScript lo hace estar presente en los navegadores más utilizados. Esto hace que el índice de penetración sea casi del 100% (Stat Owl, 2011).

- **Flash:** Es un entorno de desarrollo que permite la creación, manipulación y animación de gráficos vectoriales y rasterizados. Permite generar contenido interactivo, video y sonido. Requiere que los navegadores tengan instalado el reproductor de Flash. Esta solución, para el desarrollo de aplicaciones enriquecidas, es la más veterana, su primera versión fue lanzada en 1996, no obstante, su popularidad en el entorno web creció a partir de su cuarta versión en 1999. Su índice de penetración es algo menor que la tecnología AJAX debido a que los navegadores no dan soporte por defecto, siendo el usuario el encargado de instalar el reproductor.
- **Silverlight:** Su primera versión tuvo lugar en septiembre de 2007. Permite el desarrollo de aplicaciones RIA basadas en las capacidades de Windows Presentation Foundation y otras de .NET, como por ejemplo, funciones multimedia, animaciones e interactividad con gráficos vectoriales.
- **Flex:** Lanzado en el año 2004 por Macromedia, es un conjunto de tecnologías que permiten el desarrollo de aplicaciones RIA basadas en Flash. Flex propone un flujo de desarrollo basado en la capa de presentación. Dispone de componentes predefinidos (formularios, botones, controles...) y estilos para definir la interfaz de usuario y acciones para añadir interactividad y comportamiento dinámico.
- **Java FX:** Es una familia de productos lanzada en 2009 por Sun Microsystems para el desarrollo de aplicaciones RIA, no sólo para el ámbito web sino también para otros dispositivos como móviles, consolas de videojuegos y televisión. JavaFX es compilado a código Java por lo que puede ser ejecutado en cualquier cliente que disponga de la máquina virtual de Java.
- **Google Web Toolkit:** También denominado por sus siglas **GWT** es un kit de desarrollo de aplicaciones web creado por Google. Permite desarrollar aplicaciones RIA usando el lenguaje Java.

## 4.4 Soluciones elegidas para la implementación de Usability Planner.

Usability Planner inicialmente se planteó como una aplicación de escritorio abierta a cualquier usuario pero dirigida a un público muy específico, profesionales de la usabilidad. No hubo ninguna razón de peso para elegir una solución de escritorio en lugar de una solución web. Cualquiera de las dos encajaba con los requisitos, no obstante, una aplicación de escritorio, debido al carácter profesional y el contexto en el que se iba a utilizar fue la elegida.

Se optó por Java como lenguaje de desarrollo por su carácter universal, su independencia de la plataforma y la alta mantenibilidad del código. Para el desarrollo de la interfaz de usuario se eligió la biblioteca Swing por ser un *framework* que sigue el patrón de diseño MVC.

Se comenzó el desarrollo realizando un subconjunto del sistema, dando lugar a un primer prototipo: UCDPicker que, como se vio en el capítulo anterior, sirvió como prueba conceptual de la aplicación.

Tras replantear el alcance de la herramienta se tomó la decisión de abandonar la solución de escritorio y continuar con un desarrollo web. En ese punto surgieron nuevos requisitos:

- Reutilizar, en la medida de lo posible, el diseño realizado para el primer prototipo.
- Poder utilizar la aplicación sin estar conectado a Internet. No depender de servidores remotos ni sistemas externos.
- Guardar la información en una base de datos que se pudiera manejar sin tener conocimientos técnicos y de forma local.
- Licencia: Se requiere una licencia *opensource*.
- Portabilidad, en el sentido de no depender de sistemas externos para su funcionamiento en modo local sin conexión.
- Reutilización de código. Reutilizar el código en Java generado para el primer prototipo.
- Esfuerzo de implementación: Medido en horas estimadas de desarrollo.



En base a estos requisitos se elaboró la siguiente tabla:

Tabla 8. Comparativa de tecnologías de desarrollo.

Tecnología	Licencia	Portabilidad	Reutilización	Esfuerzo
<b>PHP</b>	Opensource	✗	★	\$
<b>JSP</b>	Opensource	✗	★★	\$\$
<b>STRUTS</b>	Opensource	✗	★★	\$\$
<b>JSF</b>	Opensource	✗	★★	\$\$
<b>ASP.NET</b>	Propietario	✗	★	\$\$
<b>Ruby On Rails</b>	Opensource	✗	★	\$\$
<b>AJAX</b>	Opensource	✓	★★	\$\$\$
<b>Flash</b>	Propietario	✓	★	\$\$
<b>Silverlight</b>	Propietario	✓	★	\$\$\$
<b>Flex</b>	Opensource	✓	★	\$\$
<b>JavaFX</b>	Opensource <sup>6</sup>	✓	★★	\$\$
<b>GWT</b>	Opensource	✓	★★★★	\$\$
<b>MySQL</b>	Opensource <sup>7</sup>	✗	-	\$\$
<b>SQL Server Express</b>	Propietario	✗	-	\$\$
<b>XML</b>	Opensource	✓	-	\$

Leyenda: ★ baja reutilización, ★★ reutilización media, ★★★ alta reutilización, \$ bajo esfuerzo, \$\$ esfuerzo medio, \$\$\$ esfuerzo alto.

Fruto del estudio realizado y en vista de los resultados de la tabla anterior se eligió **GWT** para el desarrollo del sistema y **XML** para el almacenamiento e intercambio de datos. En los siguientes dos apartados se explican en profundidad las características de ambas tecnologías.

---

<sup>6</sup> No en su totalidad.

<sup>7</sup> Sólo la version para uso no comercial.

## 4.5 GWT (Google Web Toolkit).

GWT es un *framework* de desarrollo de aplicaciones web complejas creado por Google que permite desarrollar aplicaciones RIA usando el lenguaje Java. Su potencia radica en que traduce el código Java generado en código AJAX. Está presente en las propias aplicaciones web de Google como Google Adwords<sup>8</sup>, Google Moderator<sup>9</sup> o Orkut<sup>10</sup>. Es de código abierto y completamente gratuito.

El *framework* contiene:

- **Kit de desarrollo de software**, o SDK (del inglés, *Software Development Kit*), que contiene una serie de librerías o API (Interfaz de Programación de Aplicaciones, del inglés *Application Programming Interface*), un compilador y un servidor de desarrollo. Permite la creación de aplicaciones en el lado del cliente en Java y su posterior despliegue en aplicaciones JavaScript.
- Medidor de rendimiento llamado **Speed Tracer**. Se trata de una extensión para el navegador Google Chrome que permite medir el rendimiento de las aplicaciones construidas.
- Una **extensión** para el entorno de desarrollo Eclipse para facilitar el desarrollo.
- **GWT Designer**. Una extensión para el diseño de interfaces de usuario con autogeneración de código (ver Fig. 60).

---

<sup>8</sup> <https://adwords.google.es/>

<sup>9</sup> <http://www.google.com/moderator/>

<sup>10</sup> <http://www.orkut.com/>

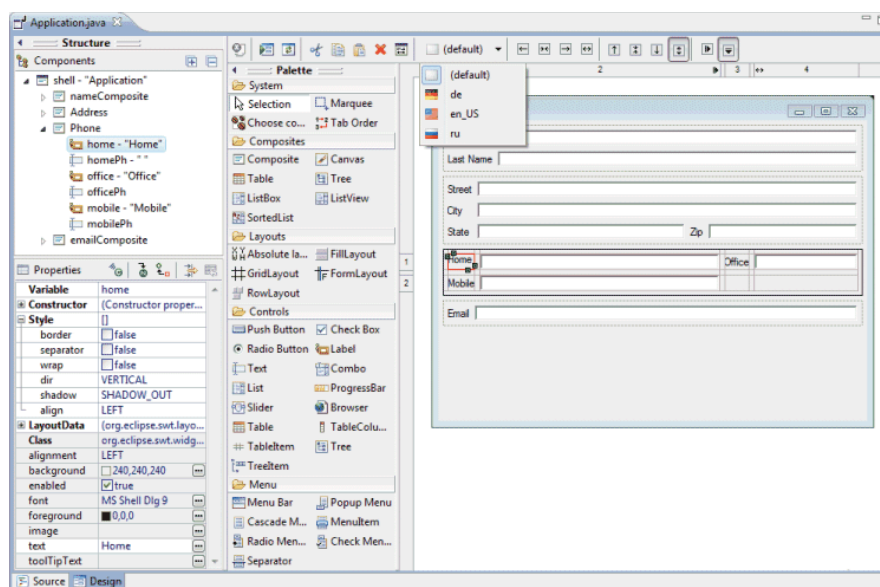


Fig. 60. GWT Designer. Generador de interfaces de usuario.

El desarrollo de una aplicación web con GWT se puede resumir en cuatro fases:

1. **Desarrollo.** El SDK de GWT proporciona un conjunto de librerías y *widgets* mediante los cuales poder diseñar y programar aplicaciones en Java para su compilación posterior en JavaScript, altamente optimizado, compatible con los principales navegadores, incluyendo dispositivos móviles. El desarrollador puede diseñar y construir la aplicación como si de una aplicación de escritorio se tratase, GWT se encarga de traducir el código Java en código JavaScript para su puesta en producción en la web.
2. **Depuración.** Se realiza en el propio código Java y en caliente, pudiendo establecer puntos de ruptura y observar valores que toman las distintas variables que contiene el programa. Además, la depuración es compatible con las herramientas propias de depuración de Java.

3. **Optimización.** GWT dispone de dos herramientas para la optimización de las aplicaciones generadas. La primera el propio compilador de GWT, el cual realiza una optimización del código JavaScript que genera, por ejemplo eliminando código muerto u optimizando las cadenas de caracteres. Además proporciona una herramienta llamada Speed Tracer, que permite diagnosticar en tiempo real problemas de rendimiento de las aplicaciones cuando se ejecutan en el navegador.
4. **Ejecución.** Cuando la aplicación se considera terminada, el resultado de la compilación es un conjunto de archivos JavaScript, HTML y CSS compatible con la mayoría de navegadores y dispositivos móviles del mercado.

De los puntos anteriores, se puede extraer que la potencia y atractivo que tiene esta plataforma de desarrollo es la posibilidad de crear aplicaciones RIA utilizando únicamente el lenguaje Java.

GWT se encarga de manejar toda la comunicación cliente-servidor mediante el uso de JSON (del inglés JavaScript Object Notation), XML, o RPCs (llamadas a procedimientos remotos), todas ellas tecnologías de intercambio de datos. El desarrollador no tiene que conocer las particularidades de cada una, simplemente utilizar la API de GWT para manejarlas.

Otras características interesantes de GWT son:

- Modo *host*: Posibilidad de desarrollar el sistema y ver los resultados sin necesidad de compilación. La aplicación se ejecuta como código de Java dentro de la Máquina Virtual de Java (JVM). Soporta el cambio de código en caliente y el depurado.
- Capacidad para crear componente reutilizables para distintos proyectos.
- Separación de la capa de presentación del modelo. GWT nos permite manejar la capa de presentación directamente en HTML o bien mediante la inclusión de componentes (o *widgets*) de su librería de componentes de interfaz de usuario.

- Soporte para el manejo del botón atrás del navegador. Este es uno de puntos débiles de muchas tecnologías RIA, las cuales no están estructuradas en páginas, causando un problema de usabilidad importante para los usuarios. GWT permite organizar la aplicación en páginas de forma que el usuario pueda interactuar con la aplicación como si de una página web se tratara, permitiendo retroceder y avanzar por el histórico de páginas que se generan.
- Internacionalización. GWT permite crear aplicaciones en múltiples idiomas, encapsulando los mensajes de la interfaz en distintos idiomas que se muestran en función del elegido por el usuario.
- Servidor. Google provee un servidor de aplicaciones para poner en funcionamiento las aplicaciones desarrolladas con GWT. Lo llama *App Engine* y permite desplegar cualquier aplicación desarrollada con GWT que requiera un servidor de aplicaciones.
- Licencia. GWT se distribuye bajo licencia Apache 2.0, la cual permite utilizarlo para crear sistemas bajo la licencia que se requiera.

## 4.6 XML.

Existen numerosas alternativas para guardar y recuperar datos en aplicaciones Web pero no todas ellas se ajustan a los requisitos del proyecto. Así por ejemplo, sistemas de gestión de bases de datos que dependen de un motor remoto para su procesamiento quedaban descartados al no permitir la ejecución de la aplicación de forma local. Este es el caso de gestores libres como MySQL.

Se requería encontrar una solución que maximizara la portabilidad y la usabilidad, es decir, facilitara la actualización de los datos de forma sencilla y sin necesidad de conocimientos avanzados en gestión de bases de datos.

En este sentido el lenguaje de marcado XML fue la opción elegida por encajar con los planteamientos de portabilidad, escalabilidad, usabilidad, costo y distribución:

- Portabilidad: Un archivo XML puede editarse con cualquier editor de texto en cualquier dispositivo y no depende de ningún servidor ni sistema para su interpretación.

- Escalabilidad: Al tratarse de un archivo con un esquema de marcado por niveles, siempre es posible su edición para incorporar niveles nuevos o modificar la información existente.
- Usabilidad: Leer y editar un archivo XML por una persona es una tarea sencilla. Permite modificar los datos “al vuelo”, sin necesidad de utilizar ningún sistema de gestión, ni compilar la aplicación. Solamente es necesario un editor de texto plano. Además la arquitectura de la información realizada encaja con la estructura de marcado con etiquetas y atributos de este lenguaje.
- Costo: Ninguno.
- Distribución: Al tratarse de un archivo, cualquiera puede editarlo según sus necesidades. Esto le otorga al sistema de mucha potencia, ya que las organizaciones o usuarios que deseen incorporar o modificar la base de conocimientos podrían hacerlo sin necesidad de recompilar, creando así versiones a medida del propio sistema.

La unión de GWT con XML fue la solución tecnológica adoptada para la implementación de la aplicación. Por un lado GWT encajaba muy bien con los requisitos del proyecto y permitía continuar el desarrollo ya realizado, por otro lado XML se eligió por ser un formato sencillo, que servía tanto para el almacenamiento de la base de datos, como para el intercambio de información y de igual forma permitía la manipulación de los datos de una forma ágil y sin necesidad de conocimientos avanzados de gestión de bases de datos.

## Capítulo 5

# DESARROLLO DEL SISTEMA.

En este capítulo se describe cómo ha sido la implementación del sistema. Se presentará su arquitectura y se explicará cada una de las partes que la integran.

Usability Planner ha sido diseñado como un sistema orientado a objetos cuya implementación se ha realizado siguiendo el patrón de diseño Modelo Vista Presentador (en adelante MVP), el cual será también presentado.

El enfoque DCU ha permitido que el sistema crezca iteración tras iteración. En cada una de estas iteraciones se ha obtenido una parte totalmente funcional del mismo; el refinamiento progresivo de los requisitos, así como los resultados de las pruebas de usabilidad han ido modelando el desarrollo hasta llegar a una versión 1.0 conforme a los requisitos funcionales y de usabilidad requeridos.

La solución que se presenta a continuación, no se considera final, pero sí se considera completa. Cumple con todos los requisitos que hasta el momento se han trazado y satisface el nivel de usabilidad planteado. No obstante, ha sido implementado de forma que pueda evolucionar y enriquecerse con nuevas características y funcionalidades.

Para hacer tecnológicamente posible este crecimiento y evolución se han utilizado tecnologías de desarrollo presentadas en el capítulo anterior, como Java, GWT, XHTML y XML; tecnologías que permiten que el ciclo de vida no se estanque, garantizando la escalabilidad y progresión del sistema.

A continuación, se detallan todas las decisiones que a nivel de implementación se han tomado para el desarrollo de Usability Planner.

## 5.1 Capas arquitectónicas.

Usability Planner ha sido diseñado como un sistema orientado a objetos dividido en varias capas arquitectónicas o subsistemas (ver Fig. 61).



Fig. 61 Capas de la arquitectura del sistema.

Se distinguen 3 capas:

- **Interfaz de usuario.** Usability Planner es accesible desde cualquier navegador con JavaScript habilitado, independientemente de la plataforma y el dispositivo. Para hacer realidad el diseño planteado se optó por el uso de hojas de estilos en cascada (o CSS), XHTML y componentes o *widgets* proporcionados por la librería de construcción de interfaces de GWT.



- **Lógica de negocio.** En esta capa están representados los objetos del dominio, en el caso de Usability Planner los métodos o técnicas de usabilidad, las etapas del ciclo de vida, las restricciones asociadas al proyecto, etc. y todas aquellas operaciones que resuelven las peticiones que el usuario realiza desde la capa superior.
- **Capa de servicios.** Residente en el servidor, contiene la base de datos, los textos de la interfaz de usuario y los servicios que GWT brinda para realizar el acceso a los datos.

GWT ofrece la cohesión y comunicación necesaria para que las 3 capas interactúen entre sí. A continuación, se presenta con más detalle cada uno de los niveles o capas que tiene la arquitectura del sistema.

#### 5.1.1 Interfaz de usuario.

La interfaz de usuario de Usability Planner está totalmente desacoplada y no contiene código que ejecute o procese la lógica de la aplicación. La función de esta capa es proporcionar la interfaz de usuario y atender y transmitir, a la capa inferior, las peticiones que el usuario realice, a la vez que mostrar los resultados obtenidos.

La interfaz ha sido construida mediante la librería de componentes o *widgets* que GWT incorpora y que permite implementar cualquier elemento que pueda representarse mediante HTML: Paneles, selectores, tablas, pestañas, botones, calendarios... (GWT Widget List, 2011). Al compilar la aplicación éstos son renderizados a HTML compatible con todos los navegadores. La maquetación y aspecto final se ha obtenido mediante hojas de estilos o CSS.

La capa de presentación de Usability Planner ha sido desarrollada de forma que pueda ser intercambiada por otra de forma inmediata. Todas las clases han sido provistas de una interfaz de manera que, si en un futuro se decidiera crear una interfaz en HTML5 o bien para un determinado dispositivo móvil, se pudiera realizar de forma totalmente transparente a la lógica de negocio.

La interacción del usuario con los distintos elementos de la interfaz, así como todas las peticiones que éste realice, son resueltas en la capa de lógica de negocio.

### 5.1.2 Lógica de negocio.

Todos los cálculos y las operaciones que Usability Planner realiza para inferir la solución y dar respuesta a las peticiones de los usuarios se resuelven en esta capa. Al tratarse de un sistema orientado a objetos implementado en Java y no depender de otras arquitecturas, su portabilidad a otras plataformas o *frameworks* como Struts, sería una operación sencilla.

Los objetos del dominio, que están presentes en esta capa, son las etapas del ciclo de vida, las restricciones asociadas al proyecto, los métodos y técnicas de usabilidad, las buenas prácticas, los beneficios y riesgos, así como las operaciones que resuelven los requisitos del sistema.

Esta capa contiene, además, operaciones para el manejo del subsistema de datos de la capa de servicios.

### 5.1.3 Capa de servicios.

Es la parte del sistema que reside en el servidor y donde está la base de conocimiento de la aplicación. Esta formada por tres elementos:

- **Base de datos.** Implementada en lenguaje XML, en ella se encuentra representada toda la base de conocimiento del sistema. Como se explicó en el capítulo 4 modificarla es tan sencillo como editar el archivo XML.
- **Idioma de la interfaz.** Todos los mensajes o textos que se muestran en la interfaz de usuario están almacenados en un archivo *.properties*. Esto permite tener varios archivos, uno por cada idioma que el sistema implemente. En la versión actual de Usability Planner, se ha incluido sólo un idioma, el inglés.
- **Servicios de GWT para acceder a los datos.** GWT dispone una serie de servicios para la lectura y escritura de datos, de forma asíncrona o síncrona, de distintas fuentes (entre ellas archivos XML).

La capa de servicios reside en el servidor de aplicaciones, de forma que cuando un usuario accede a la página de Usability Planner, se lanza una petición al servidor para que descargue los datos en el navegador del usuario, así como los textos de la interfaz. Esta operación se realiza de forma síncrona, es decir, el usuario entra en la página de la aplicación, el servidor descarga la base de datos, y hasta que no está completamente descargada no se continúa con la ejecución.

Esta solución, que puede parecer poco flexible, tiene sentido ya que Usability Planner ha sido diseñado para que pueda ser utilizado sin conexión, es decir, cualquier usuario puede descargarse en su equipo la aplicación y utilizarla sin necesidad de conectar con una base de datos remota.

## 5.2 El patrón MVP.

Google recomienda el uso de este patrón para el desarrollo de aplicaciones web bajo GWT (GWT MVP architecture, 2011). Surgió en el año 1990 en IBM y fue introducido debido a la falta de un componente intermedio en el patrón Modelo Vista Controlador (en adelante MVC) que se ocupara de la lógica de presentación (Garofalo, 2011).

Su objetivo es separar la interfaz o vista de las aplicaciones. La vista y el modelo de datos están totalmente separados. Cada vista es accedida mediante una interfaz por el presentador de modo polimórfico.

Se basa en dos características fundamentales:

- La vista y el modelo no son accesibles entre sí.
- Los presentadores son independientes de la tecnología usada para mostrar la interfaz de usuario.

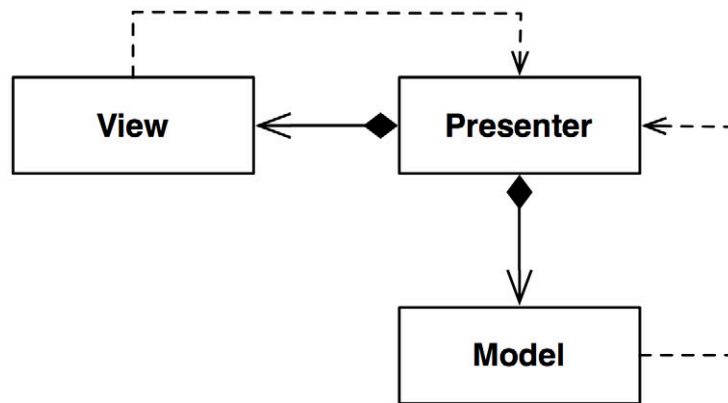


Fig. 62 MVP con vista pasiva.

Como se puede observar en la Fig. 62 el patrón tiene tres componentes:

- **La vista.** Compuesta por las clases que definen la interfaz de usuario de la aplicación.
- **El modelo de datos.** Se encarga de realizar toda la lógica de negocio sin tener ninguna información de cómo se muestra la información en la vista.
- **El presentador.** Escucha los eventos que se producen en la vista. Lleva a cabo las acciones necesarias a través del modelo. Tiene además acceso a las vistas a través de las interfaces que éstas implementan.

Existen variaciones en cuanto al grado de control sobre la vista. En Usability Planner se ha tomado la variación que Martin Fowler denominada *Vista Pasiva* (Fowler, 2006). Consiste en definir la lógica de cómo la información es formateada y presentada al usuario a partir de los controles que contiene, relegando al presentador toda la lógica de los eventos que en ella suceden.

La vista pasiva permite un mayor grado de testeabilidad ya que la lógica representada por la interfaz se reduce al mínimo, sin embargo, se aumenta la complejidad en el presentador.

### 5.2.1 Funcionamiento del presentador.

El presentador, pieza clave de este patrón, ubicado entre la vista y el modelo actúa de la siguiente forma:

1. Recibe los eventos que se producen en la vista. Por ejemplo, cuando el usuario pulsa un botón.
2. Identifica y recoge el evento, y lo convierte en una acción a ejecutar en el sistema. Por ejemplo, mostrar un mensaje. Dicha acción es resuelta en el modelo y enviada de nuevo al presentador. Por ejemplo, el modelo extrae el mensaje a mostrar de un archivo XML.
3. Devuelve el mensaje a la vista para que lo presente con un determinado formato.

El presentador queda definido por los siguientes 3 métodos:

- Un método de inicialización de la vista.
- Un método de actualización de la vista.
- Un conjunto de métodos que ejecutan operaciones sobre el modelo de datos.

Este patrón se presta mejor para aplicaciones RIA, donde, si la salida es muy compleja, se pueden implementar múltiples presentadores (Posadas, 2008).

### 5.2.2 MVP vs MVC.

Como se adelantaba al comienzo de esta sección, MVP es una variación de MVC. La diferencia principal entre ambos es el rol de la vista.

En la Fig. 63, se presenta, en primer lugar, el patrón MVP y a continuación el patrón MVC. La primera diferencia que se extrae es que la vista en MVP está totalmente separada del modelo. En MVC la vista puede leer datos directamente del modelo, al igual que el controlador.

Otra notable diferencia es el flujo de llamadas. En MVP el flujo comienza en la vista, mientras que en MVC comienza en el controlador. En MVP la vista se crea en primer lugar. Los eventos de la vista son capturados y manejados por el presentador. El resultado es devuelto por el presentador de nuevo a la vista. Por el contrario, en MVC el controlador recoge los datos del modelo, selecciona y carga la vista apropiada.

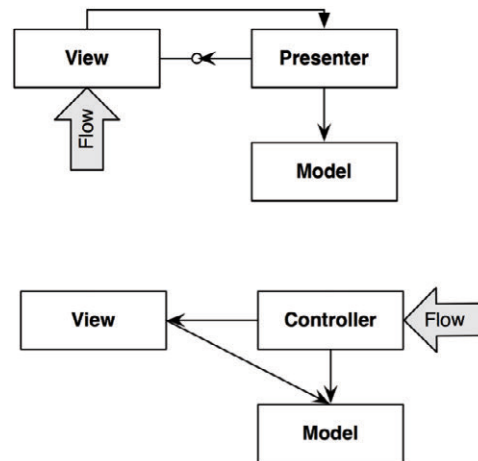


Fig. 63 MVP vs MVC

Una de las ventajas del patrón MVP es que no es necesario disponer de la vista para desarrollar el presentador, simplemente necesitamos una interfaz de cada vista. Esto permite a diseñadores y desarrolladores trabajar paralelamente.

Otra ventaja con respecto al patrón MVC, es que el presentador no tiene conocimiento de la tecnología de la interfaz de usuario que se esté utilizando. Vista y presentador conviven en clases totalmente separadas, permitiendo sustituir la vista de forma inmediata.

### 5.3 Diagrama de clases.

Una vez presentada la arquitectura del sistema y el patrón MVP, en este apartado se muestra cómo se ha implementado el diseño. Debido a las dimensiones del proyecto y al elevado número de clases que lo comprenden, se ha dividido el diseño en paquetes, de forma que pueda verse claramente la arquitectura MVP.

Una vez presentado el diagrama de paquetes (ver Fig. 64) se mostrará cómo se organizan las clases dentro de cada uno de ellos en nuevos diagramas de clase. No se han especificado todos los métodos públicos y atributos que cada clase contiene para facilitar la legibilidad de los diagramas en este documento. Por ejemplo, la mayoría de métodos públicos que escriben (métodos *set*) y leen (métodos *get*) los atributos no han sido expuestos y atributos que dependen del manejo de librerías de GWT también han sido omitidos.

### 5.3.1 Diagrama de clases (notación de paquetes).

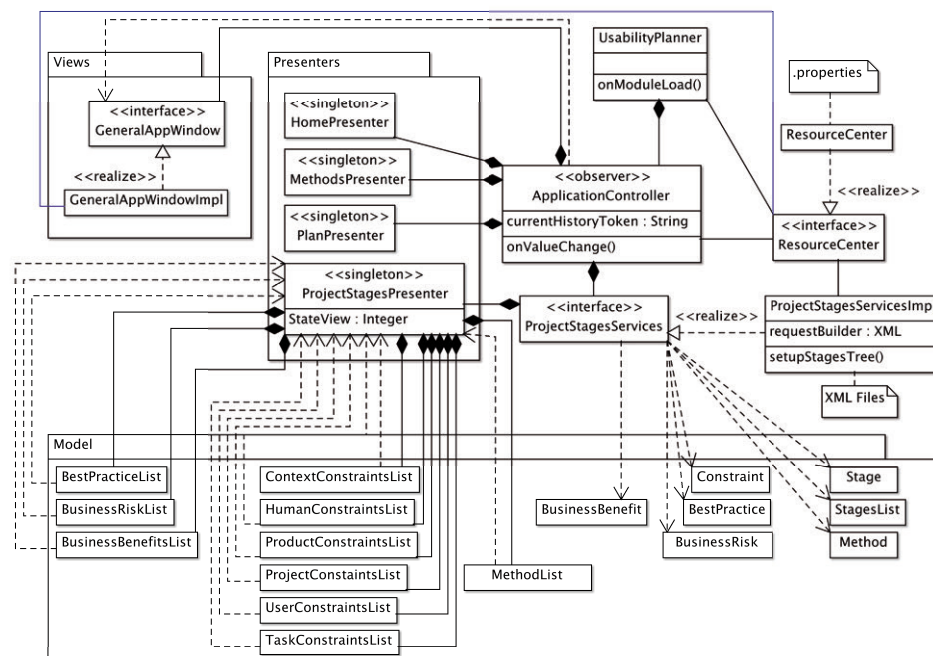


Fig. 64 Diagrama de clases (notación de paquetes).

En la Fig. 64 pueden distinguirse claramente los componentes del patrón MVP. Por un lado se tiene el paquete vistas, con las clases que implementan la interfaz de usuario, por otro lado se tiene el paquete modelo, que contiene toda la lógica de negocio y entre ambos se encuentra el paquete de los presentadores que actúan de intermediarios entre las vistas y el modelo. Las clases más relevantes son:

- Clase **UsabilityPlanner**. Es la clase principal, el origen del sistema y su papel principal es crear el controlador general de la aplicación.
- Interfaz y clase **ResourceCenter**. El papel de esta clase es proveer todos los mensajes que se muestran en la interfaz de usuario en un determinado idioma. Lee de un fichero de texto plano, con una sintaxis especial, los mensajes y los transforma en cadenas que se muestran en la vista. Todos los atributos y variables *String* del sistema están aquí representadas, por este motivo esta clase está referenciada por la mayoría de clases del sistema.
- Clase **ApplicationController**. Su misión es crear los presentadores de la aplicación, uno por cada vista, así como la clase **GeneralAppWindow** que representa los elementos comunes de la interfaz de usuario. Además se encarga de manejar los eventos de cambio de página o vista general. Cuando el usuario cambia de página, esta clase refresca o cambia a la interfaz correspondiente.
- Interfaz y clase **ProjectStagesServices**. Esta clase interactúa con la capa de servicios, para la lectura de los archivos XML que contienen la base de datos de la aplicación. Cuando el usuario accede a la página de etapas del proyecto esta clase se crea y se realiza el mapeo de la información contenida en el XML (etapas, métodos, restricciones, riesgos, beneficios y buenas prácticas) a objetos.

El resto de clases se presentarán a continuación. Se distinguen tres paquetes principales que conforman la arquitectura MVP:

- **Modelo**. El paquete modelo contiene las clases que definen la lógica de negocio de la aplicación. En el diagrama de la Fig. 64 se representan las clases que se asocian a los presentadores. En el diagrama de la Fig. 65 se expone, con más detalle, cómo están organizadas estas clases.
- **Presentadores**. Contiene las clases cuyo rol es resolver los eventos que se producen en la vista a través del modelo. En el diagrama de la Fig. 64 sólo se han representado las clases que interactúan con la vista y el modelo. En el diagrama de la Fig. 66 están representadas todas las clases que lo definen. Todas ellas son *singleton* ya que únicamente se necesita una única instancia y su creación se retrasa hasta el momento que los objetos que gobierna necesitan su control.



- **Vistas:** Contiene todas las clases que se encargan de construir la interfaz de usuario. En el diagrama de la Fig. 64 sólo se incluye la clase **GeneralAppWindow** ya que es la que construye toda la estructura común de la vista y mediante la cual el resto se comunica con los presentadores. En el diagrama de la Fig. 67 se explican el resto de clases asociadas. Todas las clases de este paquete son implementadas por interfaces con el objetivo de facilitar, en un futuro, un cambio de interfaz de usuario.

### 5.3.2 Diagrama de clases del paquete modelo.

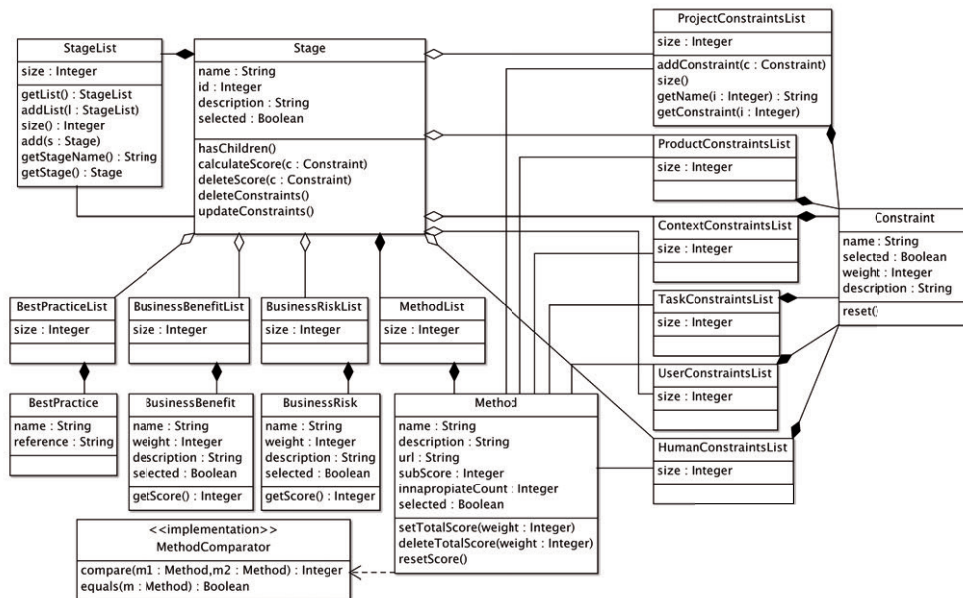


Fig. 65 Diagrama de clases (Paquete modelo).

En el diagrama de la Fig. 65 se pueden apreciar las siguientes clases:

- Clase **Stage**. Representa cada una de las etapas del ciclo de vida. Queda constituida por los atributos: Nombre que tiene la etapa (por ejemplo *Concept*), un identificador numérico, una descripción breve y un atributo que indica si la etapa ha sido seleccionada o no. A nivel de métodos, los mas representativos son:

- o `hasChildren()`. Devuelve cierto si la etapa tiene subetapas, falso en caso contrario.
  - o `calculateScore(c:Constraint)`. Para una determinada restricción `c` seleccionada, este método actualiza la puntuación de las técnicas de usabilidad o métodos que tiene recomendados.
  - o `DeleteScore(c:Constraint)`. Realiza la operación contraria a `CalculateScore`. Resta o elimina la puntuación de las técnicas cuando la restricción `c` es deseleccionada.
- Clase **StageList**. Representa listas de objetos **Stage** y se utiliza para formar las subetapas de cada etapa, así como para definir listas de etapas seleccionadas. Los métodos que contiene realizan operaciones comunes del manejo de listas, por ejemplo, calcular el tamaño, añadir o eliminar elementos, etc.
- Clase **BestPracticeList**. Es una agregación compartida de la clase **Stage**, ya que puede haber varias etapas que compartan la misma buena práctica. Sus operaciones son las comunes del manejo de listas.
- Clase **BestPractice**. Representa cada una de las buenas prácticas que pueden darse en las etapas del ciclo de vida. Queda definida por un nombre y una referencia, actualmente es el documento ISO en el que se nombra, aunque podría utilizarse en futuras versiones para proveer una URL con una descripción. Los métodos que contiene se encargan del acceso y escritura de los dos atributos anteriores.
- Clases **BusinessBenefitList** y **BusinessRisksList**. Todas las etapas tienen asociadas la misma lista de beneficios y riesgos. Sus operaciones son las comunes al manejo de listas.

- Clases **BusinessBenefit** y **BusinessRisk**. Las listas anteriores están formadas por objetos de estas clases. Ambas poseen los mismos atributos, un nombre, un peso, una descripción y un estado de selección. En la versión actual del sistema todos los beneficios y riesgos tienen la misma consideración o importancia, por lo que el peso está fijado a 1 punto para todos los objetos; no obstante, se prevé que en futuras versiones cada beneficio o riesgo pueda tener un peso distinto en cada etapa. Además de los métodos de acceso o escritura de los atributos anteriores está el método `getScore()`, el cual devuelve la puntuación de cada etapa.
- Clase **MethodList**. Todas las etapas del ciclo de vida tienen una lista de métodos recomendados. Es una parte única o composición de la clase **Stage**, ya que cada etapa tiene una lista única de métodos. Puede ocurrir que dos o mas etapas tengan una lista idéntica de métodos recomendados, pero los objetos (las técnicas de usabilidad o métodos) no son los mismos ya que cada uno tiene una puntuación distinta. Sus operaciones o métodos son los comunes al manejo de listas.
- Clase **Method**. Representa cada uno de los métodos de usabilidad que el sistema maneja. Se define mediante un nombre, una breve descripción, una URL mediante la cual mostrar una referencia o descripción más extensa, un atributo que lleva la puntuación parcial acumulada, un contador que indica la puntuación negativa y, por último, el estado de si está seleccionado o no. Por defecto todas las técnicas o métodos están seleccionados, no obstante, el usuario puede deseleccionar aquel que no le interese o no considere apropiado. Además de los métodos de acceso y escritura de los atributos anteriores se destacan:
  - o `setTotalScore(weight:Integer)`. Este método implementa el algoritmo de cálculo de la puntuación que recibe el método o técnica en función de los pesos indicados en el XML. Actualiza la puntuación parcial acumulada.
  - o `deleteTotalScore(weight:Integer)`. Realiza la operación contraria al método anterior. Actualiza la puntuación parcial acumulada cuando una restricción es deseleccionada en función de los pesos indicados en el XML.

- o `resetScore()`. Restaura las puntuaciones parciales a cero puntos.
- Clase **MethodComparator**. Es una implementación que facilita la comparación de dos objetos de clase **Method** entre sí. Se utiliza para buscar métodos en la lista de métodos.
- Clases **ProjectConstraintsList**, **ProductConstraintsList**, **ContextConstraintsList**, **TaskConstraintsList**, **UserConstraintsList** y **HumanConstraintsList**. Cada una de estas clases representa las listas de los distintos tipos de restricciones asociados a cada etapa del ciclo de vida. Cada método o técnica de usabilidad tiene además asociadas estas listas para establecer la correspondencia de los pesos por cada tipo de restricción en función de la etapa. Las operaciones que están implementadas son las comunes de manejo de listas, por ejemplo añadir o eliminar una restricción, obtener el tamaño, etc.
- Clase **Constraint**. Representa las restricciones asociadas al proyecto. Una restricción queda definida por un atributo nombre, un peso, una descripción breve y un estado de selección. Los métodos que la implementan son los habituales en la lectura y escritura de los atributos anteriores.

### 5.3.3 Diagrama de clases del paquete presentadores.

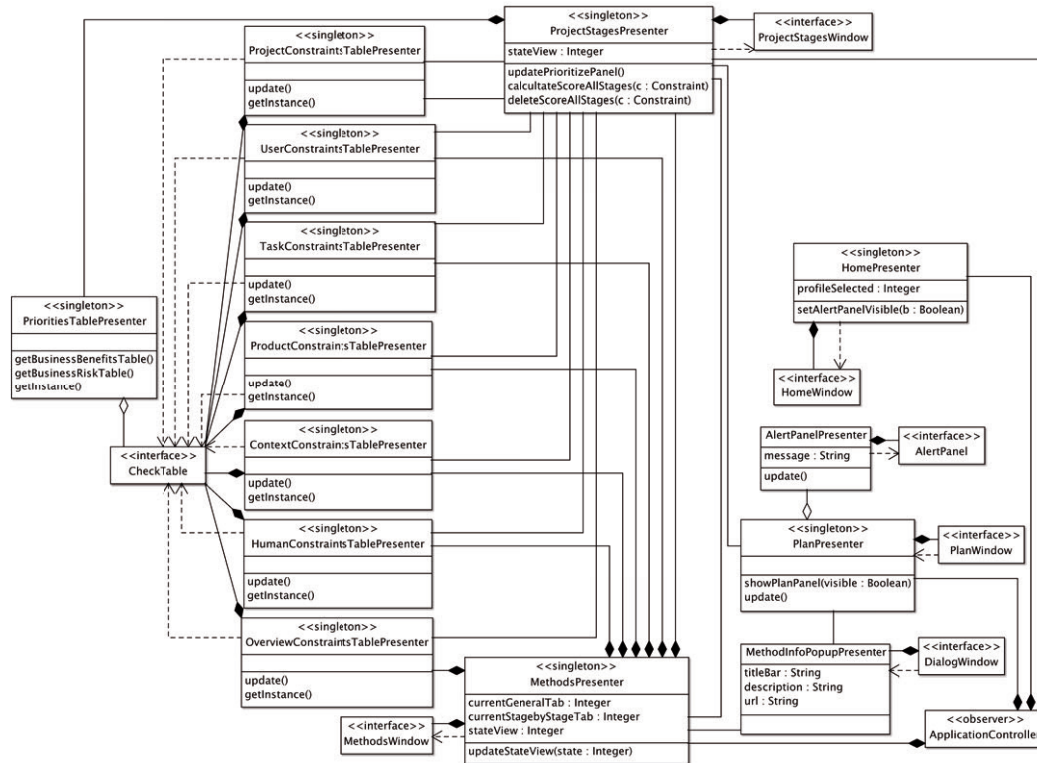


Fig. 66 Diagrama de clases (Paquete presentadores).

Como se indicaba en el apartado 5.3.1 Diagrama de clases (notación de paquetes). la mayoría de las clases que contiene este paquete son *Singleton* ya que sólo se precisa una instancia de cada una de ellas para manejar cada vista asociada. Otro aspecto importante a destacar de estas clases, es que todas son observadoras de los eventos que ocurren en las vistas. Es una característica que ofrece GWT, de forma que cuando un usuario actúa sobre la vista los presentadores resuelven las peticiones.

Las clases que contiene este paquete son:

- Clase **HomePresenter**. La responsabilidad de esta clase es atender las solicitudes que el usuario realiza desde la pantalla de inicio de la aplicación. Se encarga de pedir, bajo la demanda del usuario, el juego de etapas en función de si éste es profesional de la usabilidad, estudiante, investigador o bien si es un desarrollador. El atributo *profileSelected* adquiere un valor en función de la selección anterior.

- Clase **ProjectStagesPresenter**. Esta clase atiende a las solicitudes que el usuario realiza desde la vista de selección de etapas. En ella se manejan las listas de restricciones, buenas prácticas, riesgos, beneficios y métodos de usabilidad asociados a cada etapa del ciclo de vida. Se encarga de actualizar las listas que guardan los objetos que el usuario va seleccionando. Mediante llamadas al modelo se infieren los resultados de acuerdo a la selección del usuario y se escriben en la lista correspondiente. El atributo `stateview` indica si la ventana está mostrando la selección de etapas o bien la selección de prioridades (beneficios o riesgos). Se destacan los siguientes métodos:
  - `updatePrioritizePanel()`. Este método actualiza la vista de la tabla de riesgos y beneficios.
  - `calculateScoreAllStages(c:Constraint)`. Si un usuario selecciona una restricción que es común a todas las etapas del proyecto, este método se encarga de actualizar la puntuación de los métodos de todas las etapas del ciclo de vida seleccionadas.
  - `deleteScoreAllStages(c:Constraint)`. Si un usuario deselectiona una restricción que es común a todas las etapas del proyecto, este método se encarga de actualizar la puntuación de los métodos de todas las etapas del ciclo de vida seleccionadas.
- Clase **PrioritiesTablePresenter**. La página de selección de etapas del proyecto se divide en dos vistas, la selección de etapas y su priorización en función de los beneficios o riesgos del proyecto. Esta clase se encarga de atender a ésta última y de resolver los eventos que en ella se producen, es decir, la selección de riesgos y beneficios.
- Clase **MethodsPresenter**. Cuando el usuario ha elegido unas etapas del ciclo de vida se infieren los métodos de usabilidad recomendados. Desde la vista de métodos el usuario puede refinar la solución ajustando las restricciones que más se acercan a las características de su proyecto. Esta clase se encarga de atender las solicitudes de selección de restricciones del usuario. Los atributos `currentGeneralTab` y `currentStageByStageTab` se utilizan para saber en que pestaña se encuentra el usuario en la vista de selección por etapas o bien de selección global. Contiene un método público `updateStateView(state : Integer)` cuya misión es actualizar la vista en función de la selección del usuario.

- Clase **PlanPresenter**. Esta clase se encarga de presentar la lista final de métodos que el sistema infiere por cada etapa del ciclo de vida. Atiende a la solicitud de eliminar métodos de la lista si así el usuario lo solicitase desde la vista. Al igual que el resto de presentadores tiene un método `update()` que permite refrescar el contenido de la vista.
- Clase **MethodInfoPopupPresenter**. Esta clase construye una ventana emergente cuando el usuario pincha sobre un método, para mostrarle información relacionada. Atiende al evento cerrar ventana. Sus atributos son el título de la ventana, la información a mostrar y una URL donde el usuario puede obtener más información.
- Clase **AlertPanelPresenter**. Su papel es parecido al de la clase anterior. Se utiliza para avisar al usuario de situaciones excepcionales, por ejemplo cuando intenta visualizar el plan sin haber seleccionado ninguna etapa del proyecto.
- Clases **ProjectConstraintsTablePresenter**, **UserConstraintsTablePresenter**, **TaskConstraintsTablePresenter**, **ProductConstraintsTablePresenter**, **ContextConstraintsTablePresenter**, **HumanConstraintsTablePresenter**, **OverviewConstraintsTablePresenter**. Este conjunto de clases representan el manejo de las tablas de selección de cada tipo de restricciones. En la vista de selección de restricciones se tiene una tabla por cada tipo. Estas clases atienden a la selección de restricciones que realiza el usuario.

### 5.3.4 Diagrama de clases del paquete vistas.

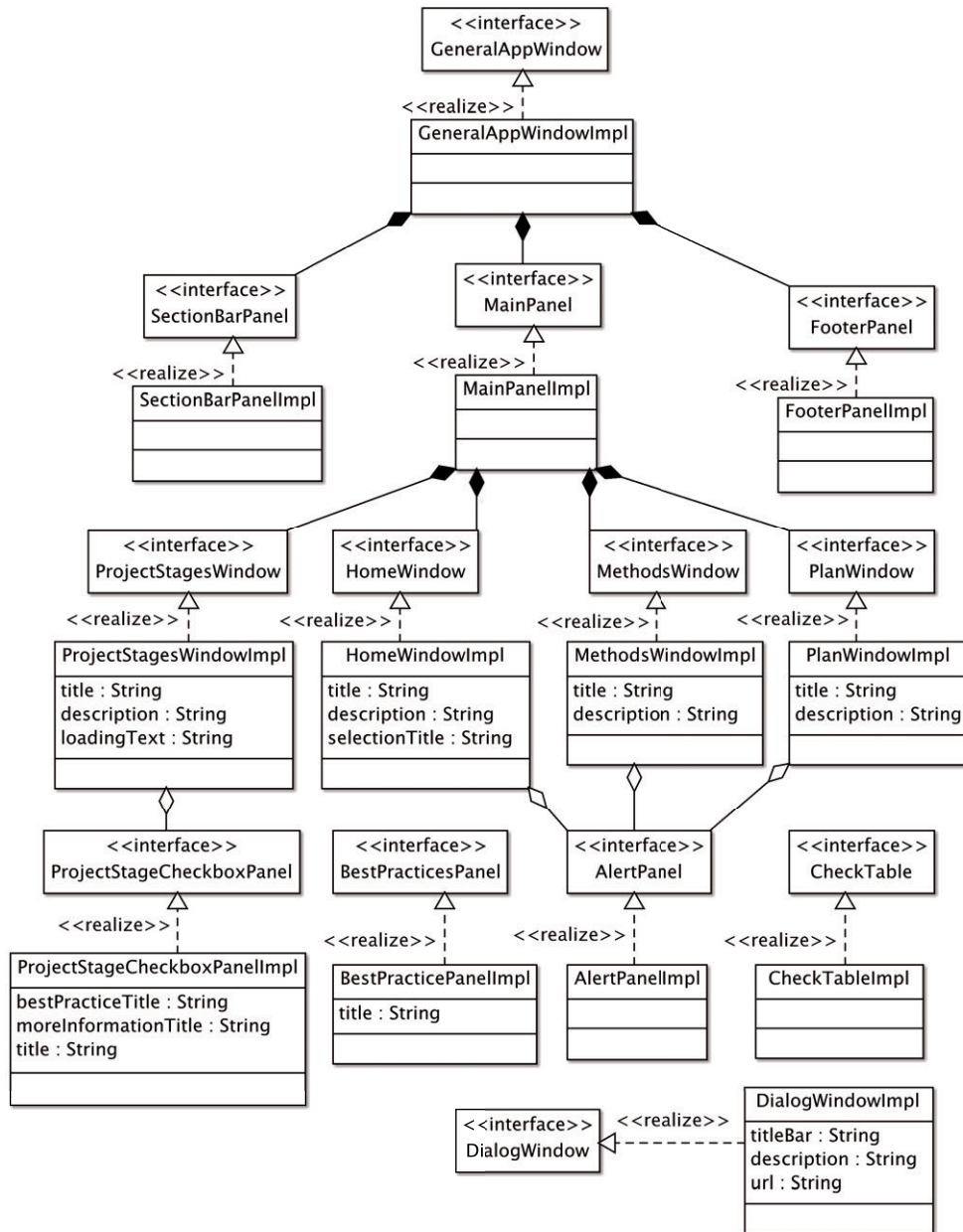


Fig. 67 Diagrama de clases (paquete vistas).



Este paquete contiene todas las clases que construyen la interfaz de usuario del sistema. Todas las vistas tienen asociadas un presentador que observa los eventos que en ellas suceden. Se destacan las siguientes clases:

- Interfaz y clase **GeneralAppWindow**. Esta clase construye la estructura general de la interfaz de usuario. Define el encabezado, el área principal, el pie, así como los botones que manejan el cambio de secciones y páginas.
- Interfaz y clase **SectionBarPanel**. Esta clase define la botonera superior que conduce a las distintas secciones de la aplicación (*Home*, *Stages*, *Methods* y *Your plan*).
- Interfaz y clase **MainPanel**. La región de la vista comprendida entre la cabecera superior y el pie se modifica en función de la sección en la que nos encontremos. Esta clase contiene la vista de la sección *Home* (implementada por la clase **HomeWindow**), la de *ProjectStages* (implementada por la clase **ProjectStagesWindow**), la de *Methods* (implementada por la clase **MethodsWindow**) y la de *Your Plan* (implementada por la clase **PlanWindow**). El controlador general de la aplicación (clase **ApplicationController**), es el encargado de intercambiar la vistas en función de la sección en la que se encuentre el usuario.
- Interfaz y clase **FooterPanel**. El pie de página contiene los botones de avance y retroceso por las secciones.
- Interfaz y clase **ProjectStageCheckboxPanel**. Cada etapa del ciclo de vida, en la ventana de selección, tiene además asociada una vista, que aparece cuando el usuario pincha sobre el enlace “más información”, en la que se describe la etapa y se muestran las buenas prácticas asociadas.

## 5.4 Diseño de la base de datos

En el capítulo anterior se analizaron distintas tecnologías válidas para la implementación de la base de datos de Usability Planner. De todas ellas el lenguaje XML es la que más encajaba con los requisitos del sistema. En esta sección se explica, mediante un diagrama entidad-relación (ver Fig. 68) cómo se han organizado los datos que maneja el sistema.

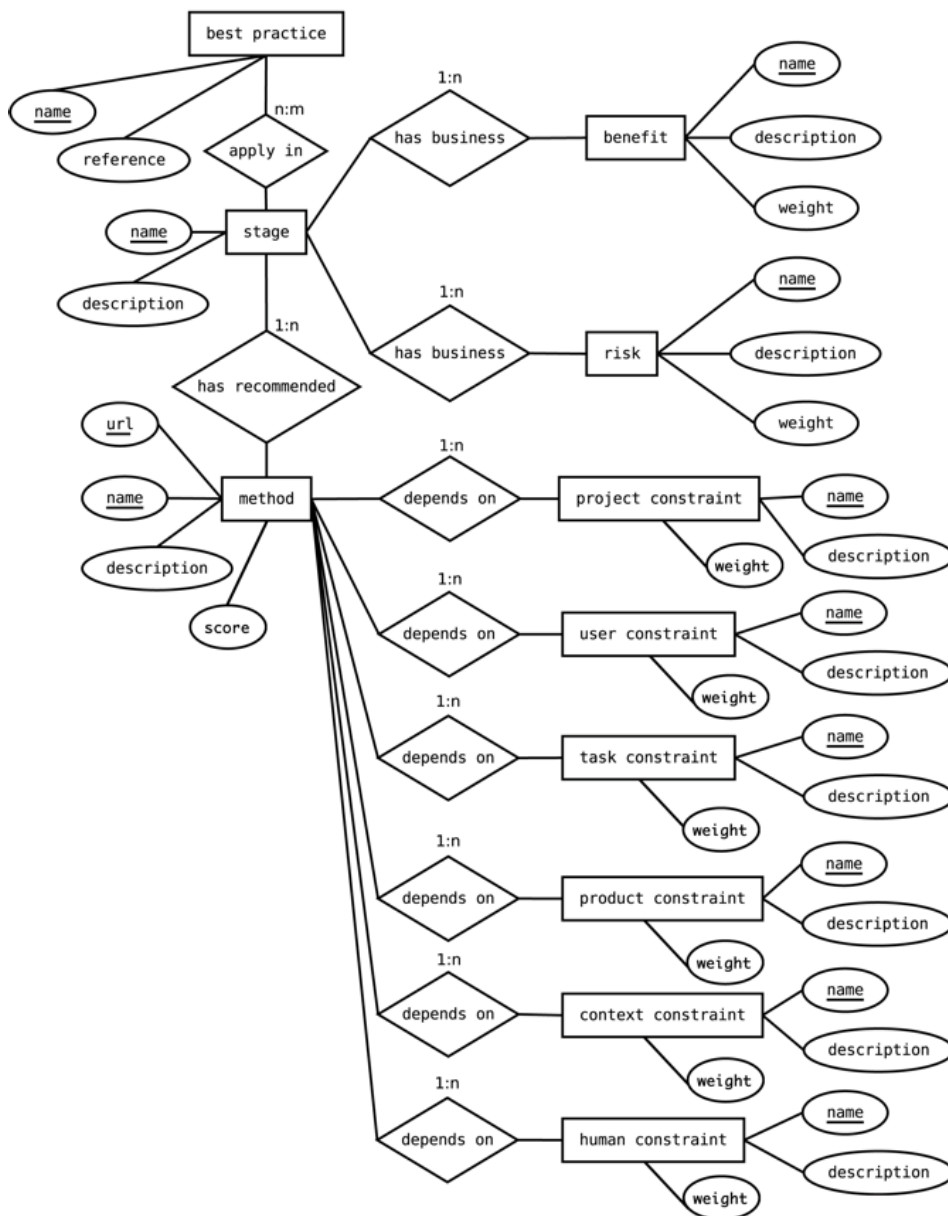


Fig. 68 Diagrama E-R del sistema.

Se pueden observar las siguientes entidades, atributos y relaciones:

- Entidad **Stage**. Representa una etapa del ciclo de vida. Tiene dos atributos, el nombre y una descripción. El nombre se identifica como clave primaria ya que no pueden existir dos etapas con el mismo nombre. Cada etapa del ciclo de vida tiene asociada riesgos y beneficios, así como métodos recomendados.
- Entidad **Best practice**. El conjunto de buenas prácticas que pueden aplicarse en las etapas del ciclo de vida quedan representadas por esta entidad. Sus atributos son el nombre (único para cada una) y una referencia al documento ISO en dónde se describe. Una buena práctica puede desempeñarse en varias etapas del ciclo de vida y viceversa, un etapa del ciclo de vida puede tener asociadas varias buenas prácticas.
- Entidades **Benefit** y **Risk**. Ambas están asociadas a cada etapa del ciclo de vida. Sus atributos son un nombre único que actúa de clave primaria, una descripción y un peso, el cual indica en qué grado se ve afectada la etapa correspondiente.
- Entidad **Method**. Cada etapa del ciclo de vida tiene una serie de métodos o técnicas de usabilidad recomendadas. Esta entidad queda descrita por el atributo nombre, url, descripción y puntuación, siendo los dos primeros claves primarias. La puntuación se obtiene en función de las restricciones del proyecto. El grado de recomendación de cada método se ve afectado por las restricciones que el usuario defina en relación con su proyecto.
- Entidades **Project constraint**, **User constraint**, **Task constraint**, **Product constraint**, **Context constraint** y **Human constraint**. Cada una de ellas representa cada tipo de restricción que el sistema maneja. Tienen dos atributos, un nombre único que actúa de clave primaria, una descripción y el peso. Este último atributo define en qué grado un método es recomendado en función de si la restricción se da en el proyecto.

## Capítulo 6

# CONCLUSIONES Y LÍNEAS FUTURAS.

Usability Planner nació con la intención de convertirse en una gran herramienta para ayudar a la comunidad de profesionales, estudiantes e investigadores, que en materia de usabilidad no tenían una vía ágil para priorizar técnicas y métodos de usabilidad minimizando los riesgos y maximizando los beneficios de negocio.

La versión actual de la herramienta cumple este objetivo y sienta las bases para la creación, en el futuro, de una aplicación mucho más potente y completa. Es la primera herramienta que resuelve el problema de la priorización de técnicas con un enfoque práctico ligado al ámbito empresarial (pero sin descuidar el ámbito académico).

Usability Planner ha sido desarrollado como proyecto de código libre, cualquiera que lo desee puede mejorarlo y adaptarlo a sus necesidades. La aplicación ha sido diseñada y desarrollada para favorecer su escalabilidad, aunque, como se explicará más adelante, parte de la selección tecnológica realizada no ha cumplido con las expectativas y puede ser un cuello de botella para futuros desarrollos.

Durante el transcurso del proyecto se han tomado decisiones que finalmente han demostrado no ser las más adecuadas y otras que sin duda habría que potenciar para mejorar la aplicación. En este capítulo se analizarán las decisiones tomadas así como los aspectos que habría que mejorar.

## 6.1 Conclusiones.

- **Complejidad existente a la hora de priorizar técnicas de usabilidad.**

Existen numerosas técnicas que permiten tratar la usabilidad durante el ciclo de vida y cada una de ellas tiene un coste. Por ejemplo, las técnicas que requieren la participación activa de los usuarios son más caras que las que no lo requieren. Sin embargo, la efectividad puede ser inversamente proporcional, la mayoría de técnicas con un coste bajo son menos efectivas que las que conllevan un coste alto.

Buscar un equilibrio entre coste y efectividad para lograr un nivel alto de usabilidad es el objetivo de toda empresa o profesional, por lo que la priorización es clave a la hora de seleccionar las técnicas a emplear.

Existen numerosas fuentes que tratan ampliamente las técnicas y buenas prácticas a realizar para asegurar la usabilidad de los productos, aunque pocas de ellas evalúan el impacto de su aplicación en un ámbito organizacional. Muchos autores, además, ofrecen visiones diferentes sobre la aplicación de algunas técnicas en los mismos escenarios.

Usability Planner surge por esta necesidad detectada, acercando a profesionales, desarrolladores e investigadores el acceso a una herramienta que permita, de forma rápida, planificar las técnicas de usabilidad más apropiadas para llevar a cabo un determinado proyecto, minimizando riesgos y maximizando beneficios.

- **Importancia del proceso de diseño iterativo y el enfoque DCU.**

El grado de incertidumbre al comienzo del proyecto sobre la solución a la que se quería llegar era bastante elevado. Se detectó una necesidad y se tomó como punto de partida una línea de investigación, pero el alcance y el resultado que se quería ofrecer fue poco a poco aclarándose gracias a la aplicación de distintas técnicas y métodos de usabilidad.

Lo primero y fundamental que todo proyecto debe contemplar es la elaboración de una estrategia para abordarlo con éxito. En el caso de Usability Planner la estrategia se basó en tres principios:

- Delimitar el problema que actualmente tienen los usuarios y organizaciones al enfrentarse a la selección y priorización de las técnicas de usabilidad. Conocer el contexto de aplicación y los estándares que tratan de ofrecer solución a este problema.
- Identificar y conocer a los potenciales usuarios de la aplicación así como sus necesidades. Crear escenarios de uso y Personas que faciliten la comprensión del problema y sirvan para comprender cómo los usuarios utilizarán la aplicación.
- Crear un prototipo de forma iterativa, con iteraciones de corta duración, y probarlo con usuarios para, iteración tras iteración, ir introduciendo mejoras.

Este último punto ha sido clave para el desarrollo de Usability Planner. El prototipado de baja fidelidad, o en papel, ha permitido probar y experimentar todo tipo de decisiones sin necesidad de implementar los cambios.

Junto con el prototipado en papel, el *feedback* de los usuarios han sido el motor que ha permitido mejorar y evolucionar la aplicación iteración tras iteración. Durante el transcurso del proyecto ha quedado demostrado que, para obtener la opinión de los usuarios, no es necesario realizar las pruebas en sofisticados laboratorios. Lo fundamental es crear tareas sencillas, fáciles de comprender y que persigan un objetivo muy concreto en el sistema. Por parte del evaluador, estar muy pendiente de cómo los usuarios las completan, cómo se desenvuelven por el sistema, que caminos toman, sus gestos y sus comentarios.

- **Importancia de buscar soluciones tecnológicas flexibles y escalables.**

Para el desarrollo de Usability Planner se buscó una solución tecnológica que permitiera implementar las necesidades y requisitos que los usuarios esperan de la aplicación, que favoreciera la flexibilidad, capacidad de respuesta y confiabilidad.

Se eligieron tecnologías basadas en Java, como Google Web Toolkit y el lenguaje de marcado XML para la gestión de los datos. Ambas tecnologías fueron elegidas tras un estudio previo y se apostó por ellas.

No obstante, tras finalizar la etapa de desarrollo y pruebas, dicha elección arrojó algún que otro problema y, sobre todo, limitaciones que en un principio no se conocían.

A pesar de ser muchas las ventajas que ofrece GWT, tras su uso se han detectado otra serie de puntos a considerar en contra de esta plataforma:

- Estabilidad. Desde el comienzo del proyecto hasta la fecha de redactar este documento se lanzaron 4 versiones importantes (desde la 2.0 a la 2.4) y otras con pequeñas modificaciones (versiones 2.X.X). Cada una incorpora nuevas características y corrige fallos detectados. Algunas funciones de librería usadas se marcaron como obsoletas de una versión a otra, produciendo inconsistencias en el código.
- Accesibilidad. Debido a que la mayor parte del resultado final es código JavaScript, construir una aplicación web accesible en GWT no es una tarea sencilla. No se tiene acceso total al DOM (Modelo de Objetos del Documento), por lo que no es posible introducir meta etiquetas y estructuras que permitan llegar a un cierto nivel de accesibilidad. No obstante, a medida de que GWT lanza nuevas versiones, incorpora nuevas características para facilitar el desarrollo de aplicaciones accesibles conforme a la especificación ARIA del organismo W3C.
- Tiempo de compilación. Desplegar el código suele llevar alrededor de un minuto. En muchas ocasiones se necesita desplegar de forma continuada para depurar errores que no son detectados en caliente.

A pesar de estos inconvenientes, GWT es una buena elección para desarrollar aplicaciones web, sobre todo si se requieren resultados rápidos.

Sin embargo, la elección de XML como lenguaje para estructurar y almacenar los datos de la aplicación no ha sido acertada. Principalmente por la escalabilidad y tiempo de acceso a los datos.

- La latencia en el acceso a los datos se ve incrementada a medida que se incrementa el volumen de información almacenada, provocando que el rendimiento de la aplicación caiga. Esto se ve reflejado en Usability Planner al comienzo, cuando el usuario accede a la aplicación. El tiempo de carga de la aplicación es elevado debido a que hay que cargar en memoria la base de datos (o la mayor parte). Esto disminuye el nivel de usabilidad, ya que la aplicación obliga al usuario a esperar unos segundos hasta que la herramienta está disponible. Los usuarios menos pacientes pueden ver un inconveniente para su uso.
- La gestión de los datos de un archivo XML también se ve comprometida a medida que se complica su estructura y crece el volumen de información. Aunque en el mercado hay editores XML muy potentes, la mantenibilidad de los datos puede ser compleja.
- Consistencia. Es muy sencillo introducir errores en el archivo, simplemente una etiqueta mal cerrada puede causar que la aplicación deje de funcionar. Localizar el error puede ser una tarea complicada si el volumen de información es elevado.

Futuras versiones de Usability Planner deberían contemplar el uso de otras tecnologías para el almacenamiento y acceso a los datos.

- **Importancia de planificar correctamente las técnicas de usabilidad a utilizar al comienzo de un proyecto.**

Es mucho mas rentable la inclusión de las actividades del Diseño Centrado en el Usuario en las etapas tempranas de un proyecto. Los métodos y técnicas de usabilidad planificados dependerán de la etapa del ciclo de vida y de restricciones relacionadas con los usuarios, las tareas a realizar en el sistema, la naturaleza del producto y la experiencia del equipo de desarrollo.

El enfoque DCU requiere más inversión en las etapas tempranas del proyecto con el objetivo de minimizar los costes durante la etapa de desarrollo. En particular, reduce el riesgo de cambios inesperados en la etapa de desarrollo.



La planificación de las técnicas y métodos de usabilidad para dar soporte al enfoque DCU debe realizarse desde el comienzo del proyecto, como parte de su planificación, y puede ser revisada en posteriores etapas del ciclo.

En las etapas tempranas de desarrollo de un proyecto las técnicas y métodos de usabilidad proveen información acerca del contexto de uso. Por ejemplo, el plan de usabilidad debe incluir la observación y entrevistas a usuarios para recopilar información sobre el contexto de uso, debe incluir el diseño colaborativo para la especificación del proceso y diseño de una solución.

En la etapa de diseño, el prototipado suele ser la técnica mas apropiada junto al cumplimiento de las guías de estilo y estándares.

En las etapas finales, cuando la implementación y las pruebas de usuario son lo más importante, la inclusión de pruebas de usuario, medidas de rendimiento y otros métodos de evaluación sirven para comprobar si los requisitos se están cumpliendo y si el grado de usabilidad alcanzado es el esperado.

## 6.2 Líneas futuras.

La escalabilidad y la capacidad para evolucionar han sido dos de los pilares del proyecto. Al tratarse de una herramienta realizada desde cero, sin la existencia de algún trabajo previo o herramienta similar, era fundamental primar un desarrollo escalable, iterativo e incremental y una metodología de trabajo que permitiera introducir cambios rápidamente.

Usability Planner se basa en un trabajo de investigación en curso, por lo que la aparición o modificación de requisitos durante el desarrollo del proyecto ha sido inevitable y natural. El enfoque de Diseño Centrado en el Usuario ha facilitado un flujo de trabajo con iteraciones muy cortas e incrementales. La restricción evaluación y las pruebas de usabilidad han servido para comprobar si la aplicación cumplía con las expectativas de los usuarios y han permitido conocer nuevas necesidades.

A continuación se detallan requisitos, funcionalidades y necesidades que han quedado fuera del alcance de la primera versión de la aplicación y que podrían incluirse en las posibles futuras versiones:

- Muchos usuarios no saben las implicaciones que tiene en la herramienta seleccionar un perfil u otro desde la pantalla de inicio. Sería más intuitivo si en la pantalla de selección de etapas del ciclo de vida se pudiese elegir el tipo de ciclo de vida que se emplea en la organización o institución en la que se encuentre el usuario (Proceso Unificado, DCU, XP...).
- La necesidad anterior abre la puerta a incluir en la aplicación más tipos de ciclos de vida. Sobre todo aquellos relacionados con metodologías ágiles que encajan muy bien con los fundamentos del diseño centrado en el usuario.
- Completar las etapas y subetapas del ciclo de vida para profesionales, estudiantes e investigadores en usabilidad. En la versión actual sólo se contempla hasta la fase de desarrollo. Lo deseable, para futuras versiones, sería que la aplicación diera soporte también para la etapa de pruebas, evaluación, monitorización, control y capacidad de mejora de la organización.
- Incluir nuevos riesgos de negocio que puedan afectar a la correcta consecución del proyecto y hacer que su selección influya en la priorización final de las técnicas de usabilidad.
- Mejorar la forma de representar la lista priorizada de técnicas. Algunos usuarios encontraron confusa la clasificación con las 5 estrellas. Sería más intuitivo dividir explícitamente la lista en los grupos propuestos por el ISO TR 16982 (recomendados, apropiados, neutrales, no recomendados y no aplicables) y dentro de cada grupo establecer un orden de mayor a menor recomendación.
- Explicar brevemente el alcance de cada restricción.
- Incluir nuevas restricciones que estén más ligadas al tipo de producto o proyecto. Muchos usuarios sugirieron, durante las pruebas, refinar los resultados en función de si el sistema a desarrollar fuese una aplicación móvil, un portal de comercio electrónico, un sistema en tiempo real, un kiosco, etc.
- Incluir un glosario completo de todas las técnicas que maneja la aplicación, con referencias bibliográficas y recomendaciones para su aplicación.

- Incluir nuevas técnicas y métodos de usabilidad.
- Guardado y recuperación del plan. Los usuarios podrían crear proyectos y guardar planes de usabilidad asociados a cada proyecto. Mediante el acceso a un área personal, el usuario podría recuperar y modificar los planes guardados.
- Exportación a formato CVS, para el tratamiento de los resultados en aplicaciones de gestión de hojas de cálculo.
- Interfaz para la gestión de la base de datos. Actualmente todos los cambios que se quieran realizar sobre la base de datos hay que realizarlos editando el fichero XML que contiene la información, y aunque no es un proceso complejo, sería más sencillo y en definitiva usable crear un pequeño gestor desde el que añadir o modificar nuevas restricciones, técnicas, etapas del ciclo de vida, buenas prácticas etc., con un pequeño historial de cambios.
- Inclusión de nuevos idiomas. La implementación del sistema se ha realizado de forma que la inclusión de nuevos idiomas sea sencillo. Simplemente creando un archivo con los mensajes y textos de la interfaz por idioma que se desee incluir.

Además de las mejoras anteriores sería muy interesante dotar a la aplicación de capacidad social. Se trataría de permitir que los usuarios expertos pudieran valorar la priorización realizada por la herramienta y que la propia herramienta tuviera capacidad para retroalimentarse y afinar los resultados en función de la valoración de los usuarios. Además, se podrían sugerir nuevas técnicas o bien proponer nuevas ordenaciones en función de las limitaciones elegidas. De esta forma el motor de inferencia podría ir perfeccionándose y basarse en el conocimiento global de la comunidad de expertos en usabilidad.



# BIBLIOGRAFÍA

Antón, T., Bevan, N., & Ferré, X. (2010). UCD Method Selection with Usability Planner. *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries* .

Bevan, N. (2009). Criteria for selecting methods in user-centred design. *I-USED* .

Bevan, N. (2007). *Determinants of Usability*. Rusia: UPA (Usability Professional Association).

Bevan, N. (2009). International Standards for Usability Should Be More Widely Used. *Journal of Usability Studies* , 106-113.

Beyer, H., & Beyer, K. (1998). *Contextual Design: Defining Customer-Centered Systems*. Morgan Kaufmann.

Cadenhead, R., & Lemay, L. (2007). *Java 6*. SAMS.

Cooper, A. (2004). *The Inmates Are Running the Asylum: Why High Tech Products Drive Us Crazy and How to Restore the Sanity (2ª Edición)*. SAMS.

Creative Commons. (2012). *Attribution-NonCommercial-ShareAlike 3.0 Unported*. From <http://creativecommons.org/licenses/by-nc-sa/3.0/>

Davis, M. E., & Phillips, J. A. (2008). *PHP y MySQL*. O'Reilly.

Ferré, X. (2005). *Marco de integración de la usabilidad en el proceso de desarrollo de software*. Facultad de Informática de la Universidad Politécnica de Madrid. Madrid: Tesis doctoral.

Ferré, X. (2000). Principios Básicos de Usabilidad para Ingenieros Software.

Ferré, X., Juristo, N., & Moreno, A. M. (2005). Framework for Integrating Usability Practices into the Software Process. *6th Int. Conf. on Product Focused Process Improvement*.

Fowler, M. (18 de Julio de 2006). *GUI Architectures*. Retrieved 25 de Enero de 2010 from Martinowler: <http://www.martinfowler.com/eaDev/uiArchs.html>

Galer, M., & Russell, A. (1987). The presentation of human factors to designers of IT products. *INTERACT'87*.

Garofalo, R. (2011). *Applied WPF 4 in Context*. Apress.

GWT MVP architecture. (2011). *Large scale application development and MVP*. <http://code.google.com/webtoolkit/articles/mvp-architecture.html>.

GWT Widget List. (2011). *Google Web Toolkit Widget List*. Retrieved Septiembre de 2011 from Google Code: <http://code.google.com/webtoolkit/doc/latest/RefWidgetGallery.html>.)

Hartson, H., & Hix, D. (1989). *Toward empirically derived methodologies and tools for human-computer interface development* (Vol. 31). International Journal of Man-Machine Studies.

Hix, D., & Hartson, H. (1993). *Developing User Interfaces: Ensuring Usability Through Product and Process*. John Wiley and Sons.

IEEE . (1990). Standard Glossary of Software Engineering Terminology.

ISO 13407. (1999). *Human-centred design processes for interactive systems*. ISO.

ISO/TR 16982. (2002). *Ergonomics of Human-System Interaction - Usability methods supporting human-centred design*. Ginebra.

Jacobson, I., Booch, G., & Rumbaugh, J. (1999). *The unified software development process*. Addison-Wesley.

Kereki, F. (2010). *Essential GWT: Building for the Web with Google Web Toolkit 2*. Addison Wesley.

Mayhew, D. J. (1999). *The usability engineering lifecycle*. San Francisco: Morgan Kaufmann Publishers.

Millet, S. (2010). *Professional ASP.NET Design Patterns*. Indianapolis: Wiley Publishing.

Nielsen Norman Group. (10 de 2011). *NNgroup*. From NNgroup: <http://www.nngroup.com/>

Nielsen, J. (1993). *Usability Engineering*. AP Professional.

Norman, D. A., & W. Draper, S. (1986). *User Centered System Design; New Perspectives on Human-Computer Interaction*. Hillsdale: L. Erlbaum Associates Inc.

ORACLE. (2010). *The Java EE 6 Tutorial*. Redwood: ORACLE.

O'Reilly, T. (2005). What is Web 2.0. Design patterns and business models for next generation of software. *O'Reilly Media*.

Posadas, M. (2008). *Programación con Silverlight 2.0*. Madrid: Netalia.

S. Presuman, R. (2005). *Ingeniería del Software: Un enfoque practico*. McGraw-Hill.

Sanders, B. (2011). *Smashing HTML5*. Wiley.

Seffah, A., & Metzker, E. (Diciembre de 2004). The obstacles and myths of usability and software engineering. *Communications of the ACM*.

Seffah, A., & Metzker, E. (Diciembre de 2004). The obstacles and myths of usability and software engineering. *Communications of the ACM* 47.

Sharma, P. (2011). *Techpluto*. Retrieved 11 de 2011 from Techpluto: [www.techpluto.com/web-20-services/](http://www.techpluto.com/web-20-services/)

Siegel, D. (1996). *Creating killer Web sites: the art of third-generation site design*. Hayden Books.

Stat Owl. (julio de 2011). *Rich Internet Application Market Share*. Retrieved 1 de julio de 2011 from Stat Owl: [http://www.statowl.com/custom\\_ria\\_market\\_penetration.php](http://www.statowl.com/custom_ria_market_penetration.php)

Stellman, A., & Greene, J. (2005). *Applied Software Project Management*. O'Reilly Media.

SWEBOK. (2004). *IEEE Computer Society*. (IEEE, Producer) From <http://www.computer.org/portal/web/swebok>

Tate, B. A., & Hibbs, C. (2007). *Ruby on Rails*. O'Reilly.

Usability Professionals Association. (2010). *Usability Body of Knowledge*. Retrieved Enero de 2012 from Usability Body of Knowledge: [www.usabilitybok.org/glossary](http://www.usabilitybok.org/glossary)

User Experience Professionals Association. (n.d.). *UXPA Projects: Usability Body of Knowledge*. Retrieved Diciembre de 2011 from [http://www.usabilityprofessionals.org/upa\\_projects/body\\_of\\_knowledge/bok-mission.html](http://www.usabilityprofessionals.org/upa_projects/body_of_knowledge/bok-mission.html)

Wixon, D. (2003). Evaluating usability methods: why the current literature fails the practitioner. *Interactions* .



# ANEXOS

1. Relación entre buenas prácticas, técnicas y etapas.
2. Etapas del ciclo de vida consideradas.
3. Cuestionario de satisfacción.

## Anexo 1

# RELACIÓN ENTRE BUENAS PRÁCTICAS, TÉCNICAS Y ETAPAS.

En el apartado 2.3, se presentaron distintas clasificaciones de técnicas y métodos de usabilidad para dar soporte a las actividades del enfoque DCU.

La clasificación basada en etapas del ciclo de vida (apartado 2.3.3) hacía mención, primero, a identificar las buenas prácticas, que en términos de usabilidad, son viables para minimizar los riesgos de proyecto, buenas prácticas identificadas por el ISO PAS 18152.

La siguiente tabla (Tabla 9) recoge ejemplos de métodos de usabilidad que dan soporte a las buenas prácticas que en materia de usabilidad minimizan riesgos.

En la primera columna se detallan las principales buenas prácticas que, en una determinada etapa del ciclo de vida (tercera columna), son necesarias cumplir para obtener un buen nivel de usabilidad. Las técnicas y métodos para llevarlas a cabo están enumeradas en la segunda columna.

La priorización de cada una de las técnicas y métodos anteriores se basa en el criterio descrito en el ISO/TR 16982.

Tabla 9. Buenas prácticas y técnicas de usabilidad para alcanzarlas por etapas del ciclo de vida.

Buenas prácticas identificadas en el ISO 18152	Técnicas y métodos de usabilidad recomendados	Etapas del ciclo de vida
<ul style="list-style-type: none"> <li>- Identificación del contexto de uso del sistema (necesidades, tendencias y expectativas).</li> <li>- Conceptualizar el sistema (aclarar objetivos, su viabilidad y riesgos).</li> </ul>	<ul style="list-style-type: none"> <li>- Reuniones de trabajo para identificar necesidades futuras.</li> <li>- Estudios de campo preliminares.</li> <li>- Grupos de enfoque.</li> <li>- Estudios con fotografías.</li> <li>- Simulaciones del contexto o entorno futuro.</li> <li>- Tormenta de ideas.</li> <li>- Tormenta de ideas (mediante dibujos).</li> <li>- Análisis en profundidad del trabajo y estilo de vida de los usuarios.</li> </ul>	1. Conceptualización.
		1.1. Prever oportunidades.
<ul style="list-style-type: none"> <li>- Describir los objetivos que los usuarios u organización de usuarios quieren cumplir mediante el uso del sistema.</li> <li>- Definir el alcance del contexto de uso del sistema.</li> </ul>	<ul style="list-style-type: none"> <li>- Reuniones de trabajo participativas.</li> <li>- Estudios de campo o etnográficos.</li> <li>- Consulta a las partes interesadas.</li> <li>- Análisis de factores humanos.</li> <li>- Análisis del contexto de uso.</li> </ul>	1.2. Alcance del sistema.
<ul style="list-style-type: none"> <li>- Decidir las responsabilidades, roles y tareas de la organización.</li> <li>- Investigar las necesidades y requisitos de usabilidad necesarios para garantizar la usabilidad en todo el ciclo de vida del sistema.</li> </ul>	<ul style="list-style-type: none"> <li>- Plan para alcanzar y mantener la usabilidad.</li> </ul>	2. Planificación.
<ul style="list-style-type: none"> <li>- Identificar y analizar los roles de cada grupo de usuarios interesados y cómo se ven afectados por el sistema.</li> <li>- Describir las características de los usuarios.</li> <li>- Describir el entorno cultural y de la organización.</li> <li>- Describir las características de cualquier agente externo al sistema y su entorno de funcionamiento.</li> <li>- Analizar las implicaciones del contexto de uso.</li> <li>- Presentar el contexto de uso a las partes interesadas.</li> </ul>	<ul style="list-style-type: none"> <li>- Identificación de partes interesadas.</li> <li>- Estudios de campo o etnográficos.</li> <li>- Reuniones de trabajo participativas.</li> <li>- Análisis del contexto de uso.</li> <li>- Análisis del contexto de trabajo.</li> <li>- Análisis de datos por eventos.</li> <li>- Definición del contexto de uso y su alcance.</li> <li>- Diarios.</li> </ul>	3. Captura de requisitos.
		3.1. Contexto.
<ul style="list-style-type: none"> <li>- Definir el presupuesto, habilidades, equipo de soporte necesario para cumplir todas las tareas.</li> </ul>	<ul style="list-style-type: none"> <li>- Análisis de tareas.</li> <li>- Análisis del contexto de trabajo.</li> </ul>	3.2. Tareas.
<ul style="list-style-type: none"> <li>- Realizar un estudio o investigación sobre las necesidades de usabilidad del sistema.</li> </ul>	<ul style="list-style-type: none"> <li>- Análisis de la usabilidad del sistema.</li> </ul>	3.3. Necesidades de usabilidad.

	<ul style="list-style-type: none"> <li>- Mediciones de usabilidad.</li> <li>- Análisis de la competencia.</li> <li>- Evaluación heurística o por expertos.</li> </ul>	
<ul style="list-style-type: none"> <li>- Generar varios diseños para cada uno de los aspectos del sistema que tienen relación con los usuarios.</li> <li>- Producir soluciones centradas en el usuario para cada opción de diseño.</li> </ul>	<ul style="list-style-type: none"> <li>- Prototipado temprano y evaluación de usabilidad</li> <li>- Simuladores.</li> <li>- Diseño en paralelo.</li> </ul>	3.4. Diseño.
<ul style="list-style-type: none"> <li>- Analizar las implicaciones del contexto de uso.</li> </ul>	<ul style="list-style-type: none"> <li>- Definición del contexto de uso y su alcance.</li> </ul>	4. Requisitos.
		4.1. Relacionados con el contexto de uso.
<ul style="list-style-type: none"> <li>- Redactar una declaración explícita de los requisitos de usuario.</li> </ul>	<ul style="list-style-type: none"> <li>- Escenarios.</li> <li>- Personas.</li> <li>- Guiones gráficos (viñetas).</li> <li>- Definición de los requisitos de la interfaz de usuario.</li> <li>- Establecer metas de rendimiento y satisfacción.</li> <li>- Priorización de requisitos.</li> </ul>	4.2. Requisitos de usuario.
<ul style="list-style-type: none"> <li>- Verificar que el diseño propuesto sigue los criterios de usabilidad</li> <li>- Analizar los requisitos de usuario.</li> <li>- Presentar los requisitos a implementar a las partes interesadas.</li> </ul>	<ul style="list-style-type: none"> <li>- Identificar y analizar los requisitos de las partes interesadas.</li> <li>- Formato estándar de la industria para la especificación de requisitos.</li> <li>- SWOT análisis.</li> <li>- QDF</li> </ul>	5. Análisis de requisitos.
<ul style="list-style-type: none"> <li>- Generar varios diseños para cada uno de los aspectos del sistema que tienen relación con los usuarios.</li> <li>- Producir soluciones centradas en el usuario para cada opción de diseño.</li> <li>- Permitir la personalización para dar soporte a las necesidades del mercado o de los usuarios.</li> <li>- Distribuir las funciones entre el sistema, el usuario y la organización.</li> <li>- Diseñar un modelo teniendo presente los requisitos, el contexto de uso y las restricciones de diseño del sistema.</li> <li>- Desarrollar una descripción de cómo el sistema va a ser utilizado para dar soporte a la integración de cada componente.</li> </ul>	<ul style="list-style-type: none"> <li>- Asignación de funciones y tareas.</li> <li>- Generar opciones de diseño.</li> <li>- Ergonomía.</li> <li>- Diseño participativo.</li> <li>- Guías de interfaz de usuario y estándares.</li> <li>- Preguntas, opciones y criterio.</li> </ul>	6. Diseño / desarrollo.
		6.1 Diseño de alto nivel.
<ul style="list-style-type: none"> <li>- Desarrollar un prototipo que simule los aspectos principales del sistema.</li> </ul>	<ul style="list-style-type: none"> <li>- Desarrollo de prototipos.</li> <li>- Desarrollo de simuladores.</li> <li>- Desarrollo de <i>wireframes</i>.</li> </ul>	6.2. Prototipado.
<ul style="list-style-type: none"> <li>- Realizar evaluaciones heurísticas o por expertos del prototipo.</li> <li>- Realizar test de usuario del prototipo.</li> </ul>	<ul style="list-style-type: none"> <li>- <i>Card sorting</i> (Arquitectura de información).</li> <li>- Análisis de peticiones.</li> <li>- Tutorial cognitivo.</li> <li>- Técnica de incidentes críticos.</li> <li>- Evaluación formativa.</li> </ul>	6.3 Evaluación de usabilidad.

	<ul style="list-style-type: none"> <li>- Evaluación sumativa.</li> <li>- Tutorial de usabilidad.</li> <li>- Evaluación remota.</li> <li>- Cuestionarios de satisfacción.</li> <li>- Mago de Oz.</li> </ul>	
<ul style="list-style-type: none"> <li>- Implementar las estrategia de recursos humanos de la organización.</li> <li>- Motivar a las personas y a los equipos para que trabajen juntos para alcanzar los objetivos de la organización.</li> </ul>	<ul style="list-style-type: none"> <li>- Análisis de dominio de trabajo.</li> <li>- Análisis de tareas.</li> <li>- Diseño participativo.</li> <li>- Evaluación de la carga de trabajo.</li> <li>- Modelos de rendimiento.</li> </ul>	6.4 Recursos humanos.
<ul style="list-style-type: none"> <li>- Definir las necesidades de personal y habilidades que se necesitan para alcanzar todas las tareas.</li> <li>- Identificar las habilidades que se necesitan para cada rol.</li> <li>- Identificar necesidades de personal futuras.</li> <li>- Calcular los recursos disponibles en términos de horas, esfuerzo y disponibilidad.</li> </ul>		
<ul style="list-style-type: none"> <li>- Planificar la evaluación.</li> <li>- Identificar y analizar las condiciones sobre las cuales se va a realizar la evaluación.</li> <li>- Revisar que el sistema está preparado para su evaluación.</li> <li>- Revisar el diseño y características de seguridad haciendo uso de la información obtenida de las pruebas de evaluación.</li> <li>- Analizar los resultados de la evaluación.</li> </ul>	<ul style="list-style-type: none"> <li>- Obtención de la opinión de los usuarios.</li> <li>- Uso de modelos y simulación.</li> </ul>	7. Pruebas
		7.1. Planificación y ejecución.
<ul style="list-style-type: none"> <li>- Comprobar que el sistema cumple todos los requisitos del usuario.</li> <li>- Revisar que la interfaz de usuario cumple con los criterios de usabilidad, guías de estilo, estándares y con la legislación vigente.</li> </ul>	<ul style="list-style-type: none"> <li>- Validación contra los requisitos.</li> <li>- Formato estándar para la redacción de informes de usabilidad.</li> <li>- Medidas de rendimiento.</li> <li>- Guías de interfaz de usuario y estándares.</li> </ul>	7.2. Validación

## Anexo 2

# ETAPAS DEL CICLO DE VIDA CONSIDERADAS.

En el apartado 2.3.3 se hace mención a las etapas del ciclo de vida que maneja Usability Planner para perfiles de usuario ligados profesionalmente a la usabilidad (profesionales, estudiantes e investigadores) y para perfiles más ligados al desarrollo. A continuación se describe cada una de las etapas y sub etapas de cada ciclo de vida consideradas en la herramienta.

### A.2.1 Actividades o etapas del ciclo de vida del enfoque DCU.

1. **Conceptualización:** Etapa en la que se identifica o conceptualiza el sistema.
  - 1.1. **Identificar oportunidades:** Sub-etapa en la que se estudian las posibles oportunidades de negocio para crear o mejorar un sistema.
  - 1.2. **Alcance:** En esta sub-etapa se define los objetivos que el sistema debe cubrir y sus casos de uso.
2. **Planificación:** En esta etapa se definen las técnicas y métodos de usabilidad que se deben utilizar para asegurar la consecución de las buenas prácticas.
3. **Identificar necesidades:** Etapa en la que se definen los usuarios a los que va dirigido el sistema así como sus necesidades.
  - 3.1. **Contexto de uso:** En esta sub-etapa se estudia e identifica el contexto u entorno en el que los usuarios van a utilizar el sistema.
  - 3.2. **Análisis de tareas:** Sub-etapa en la que se identifican las tareas que el sistema va a realizar.
  - 3.3. **Necesidades de usabilidad,** en términos de rendimiento y satisfacción de los usuarios.
  - 3.4. **Opciones de diseño:** Sub-etapa en la que se analizan posibles soluciones de diseño de cada aspecto del sistema.
4. **Requisitos:** Etapa en la que se definen los requisitos del sistema.
  - 4.1. **Contexto de uso:** En esta sub-etapa se especifica el contexto u entorno en el que los usuarios van a utilizar el sistema.
  - 4.2. **Requisitos de usuario:** En esta sub-etapa se especifican los requisitos de usuario.
5. **Análisis de los requisitos:** En esta etapa se analiza si realmente los requisitos especificados alcanzan las necesidades de los usuarios.

6. **Diseño y desarrollo:** Etapa dedicada al diseño, desarrollo y evaluación de soluciones.
  - 6.1. **Diseño de alto nivel:** Sub-etapa en la que se diseña como el sistema va a funcionar.
  - 6.2. **Prototipado:** En esta sub-etapa se implementan prototipos para su evaluación.
  - 6.3. **Evaluación de usabilidad:** En esta sub-etapa se debe comprobar si el nivel de usabilidad alcanzado es el esperado.
  - 6.4. **Recursos humanos:** En esta sub-etapa se evalúa el grado en el que el sistema influye en los recursos de la organización (personas involucradas, habilidades, roles, formación...).

#### A.2.2 Etapas del ciclo de vida del Proceso Unificado de Rational.

- **Inicio:** Etapa en la que se define el alcance del proyecto, se identifican las necesidades y se analiza la viabilidad y riesgos en términos de negocio. Además se define su planificación.
- **Elaboración:** Etapa en la que se definen los casos de uso que permiten definir la arquitectura del sistema. Se completan los requisitos y se diseña una solución preliminar.
- **Implementación:** En esta etapa se desarrolla el sistema y se aplican los cambios en función de las evaluaciones realizadas por los usuarios.
- **Cierre:** El objetivo de esta fase es asegurar que el sistema está preparado para los usuarios finales. Se ajustan los errores encontrados en las pruebas y se comprueba que el sistema cumple con los requisitos especificados.



## Anexo 3

# CUESTIONARIO DE SATISFACCIÓN.

Durante el congreso NordiCHI de 2010 se tuvo la oportunidad de realizar un test de usabilidad a profesionales del ámbito de la usabilidad. Para la preparación de la evaluación se creó una plantilla o modelo de cuestionario de satisfacción.

A continuación se incluye el modelo de cuestionario que fue entregado a cada participante.

Nombre	
Ocupación	
Académico / Industria	
¿Qué tipo de sistemas desarrollas o evalúas?	
¿Qué técnicas o métodos para <b>evaluar la usabilidad</b> usas normalmente?	
¿Qué otros métodos de usabilidad normalmente utilizas?	
¿Cuántas veces al año utilizas los métodos indicados anteriormente?	
Número de años de experiencia	
¿Enseñas o impartes cursos de usabilidad?	
¿Cuál es tu experiencia evaluando la usabilidad?	Ninguna   Limitada   Moderada   Experto
¿Cuál es tu experiencia con otros métodos?	Ninguna   Limitada   Moderada   Experto
¿En que grado utilizarías Usability Planner para ayudarte a priorizar métodos?	(Nunca) 1 2 3 4 5 6 7 (Siempre)
¿Recomendarías esta aplicación a alguien? ¿A qué tipo de persona?	(Nunca) 1 2 3 4 5 6 7 (Siempre) Tipo de persona:

**Comentarios:**

